CHINESE
JOURNAL OF ELECTRONICS

RuleMaster+: LLM-Based Automated Rule Generation Framework for Intrusion Detection Systems

Wenjuan Lian, Chengxin Zhang, Hongbao Zhang, Bin Jia, Baihang Liu

Citation: Wenjuan Lian, Chengxin Zhang, Hongbao Zhang, Bin Jia, Baihang Liu, RuleMaster+: LLM-Based Automated Rule Generation Framework for Intrusion Detection Systems, Chinese Journal of Electronics, 2025, 34(5), 1-13. doi: 10.23919/cje.2024.00.342.

View online: https://doi.org/10.23919/cje.2024.00.342

Related articles that may interest you

FlowGANAnomaly: Flow-Based Anomaly Network Intrusion Detection with Adversarial Learning
Chinese Journal of Electronics. 2024, 33(1), 58-71 https://doi.org/10.23919/cje.2022.00.173

Survey on Fake Information Generation, Dissemination and Detection
Chinese Journal of Electronics. 2024, 33(3), 573-583 https://doi.org/10.23919/cje.2022.00.362

A General Authentication and Key Agreement Framework for Industrial Control System
Chinese Journal of Electronics. 2024, 33(4), 1046-1062 https://doi.org/10.23919/cje.2023.00.192

Traffic Sign Recognition Using an Attentive Context Region-Based Detection Framework
Chinese Journal of Electronics. 2021, 30(6), 1080-1086 https://doi.org/10.1049/cje.2021.08.005

Frame Synchronization Method Based on Association Rules for CNAV-2 Messages
Chinese Journal of Electronics. 2023, 32(2), 295-302 https://doi.org/10.23919/cje.2021.00.148

Improving Knowledge Graph Completion Using Soft Rules and Adversarial Learning
Chinese Journal of Electronics. 2021, 30(4), 623-633 https://doi.org/10.1049/cje.2021.05.004

In Press

Follow CJE WeChat public account for more information

Chinese Journal of Electronics
vol. 34, no. 5, pp. 1–13, September 2025
https://doi.org/10.23919/cje.2024.00.342

Special Focus on Large Language Model Security and Privacy
RESEARCH ARTICLE

RuleMaster+: LLM-Based Automated Rule Generation Framework for Intrusion Detection Systems

Wenjuan Lian[1], Chengxin Zhang[1], Hongbao Zhang[2], Bin Jia[1], and Baihang Liu[1]

1. *College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
2. *Eversec Technology Co., Ltd., Beijing 100000, China*

Corresponding author: Bin Jia, Email: jiabin@sdust.edu.cn
Manuscript Received December 20, 2024; Accepted March 27, 2025; Published Online May 16, 2025
Copyright © 2025 Chinese Institute of Electronics

Abstract — Intrusion detection is vital for network security, but traditional intrusion dectetion system (IDS) face challenges in complex rule writing, requiring significant resources. The rise of large language models (LLMs) offers new possibilities, yet limited research exists on applying LLMs to IDS rule generation. Existing LLMs often fall short for this task. This paper presents RuleMaster, an LLM fine-tuned with a curated instruction dataset to automate IDS rule generation. Even for new attacks, as long as the proof-of-concepts (PoC) for the attack is available, the RuleMaster can generate effective IDS rules. Our evaluation shows that RuleMaster outperforms ChatGPT in rule generation tasks, as demonstrated through winrate evaluation and manual assessment. Furthermore, we refine RuleMaster into RuleMaster+, which not only generates IDS rules but also provides explanations for rules and PoC, along with corresponding defense recommendations. Our approach effectively improves the quality and efficiency of rule writing, providing a new solution for enhancing IDS performance.
Keywords — Large language model, Intrusion detection, Rule generation, Cybersecurity.
Citation — Wenjuan Lian, Chengxin Zhang, Hongbao Zhang, *et al.*, "RuleMaster+: LLM-based automated rule generation framework for intrusion detection systems," *Chinese Journal of Electronics*, vol. 34, no. 5, pp. 1–13, 2025. doi: 10.23919/cje.2024.00.342.

In Press

1. Introduction

The rapid growth of the Internet has brought convenience but also cybersecurity concerns. Conventional security technologies like access control, encryption, and firewalls are insufficient in addressing modern network security needs, as they only react to external threats. To tackle these challenges, research on advanced intrusion dectetion system (IDS) is crucial [1]. By pre-configuring rules and continuously monitoring the network, IDS can detect potential intrusions, providing timely alerts to security personnel for prompt decision-making. They can also update rules to detect evolving threats like port scanning, file uploads, structured query language (SQL) injections, and cross-site scripting attacks.

Snort, a leading open-source IDS [2], uses rules to detect malicious network activity and alert users, boasting over 5 million downloads and 600,000 users as of March 2024. Suricata, a strong alternative, addresses Snort's limitations by detecting segmented signatures across Transmission Control Protocol (TCP) packets, capturing application-layer data, and surpassing Snort in performance and functionality. While IDS solutions offer advantages like high-performance traffic analysis, fast detection, and easy deployment, evolving attack methods challenge their effectiveness, as configured rules often fail to cover all threats. This necessitates continuous rule updates, which require expertise, contextual adjustments, and significant labor, making traditional IDS solutions costly and time-consuming.

Recently, the explosive popularity of ChatGPT has led to the emergence of large language models (LLMs) that are capable of generating a variety of content on human instruction, such as writing articles, making graph-

Associate Editor: Prof. Debiao He, Wuhan University.

ics, coding, and so on, demonstrating powerful natural language capabilities [3]. However, using LLMs directly for generating IDS rules still faces challenges. Most mainstream LLMs are trained on general text data, leading to subpar performance within specific professional domains like network security [4].

Our research aims to automatically generate high-quality IDS rules by fine-tuning LLM, addressing the issues of incomplete rule sets and complex rule writing in existing IDS. Additionally, we provide comprehensive interpretation of these rules and proof-of-concepts (PoCs), along with corresponding protection recommendations. In summary, our main contributions are as follows:

1) Our research proposes a novel approach to automatically generate IDS rules by fine-tuning LLMs, reducing manual overhead and addressing issues like incomplete coverage and rule formulation difficulty. Additionally, we fine-tuned the model with an explanation instruction dataset, enabling it to generate IDS rules, provide explanations, and offer defense recommendations.

2) We developed the inaugural cybersecurity instruction dataset, specifically designed for fine-tuning LLM to automate IDS rule generation. This comprehensive dataset comprises 3002 instructions encompassing PoC and IDS rules, and is openly accessible for researchers to enhance LLM capabilities in the realm of network security. The GitHub link for our dataset and code is: https://github.com/Rulemaster215/Rulemaster.

3) Our research is capable of enhancing the detection capabilities of existing IDS by seamlessly integrating automatically generated IDS rules into current systems, enabling more precise identification and capture of network intrusion behaviors. This integration significantly enhances intrusion detection performance and overall network security.

4) Experimental results demonstrate that RuleMaster surpasses original LLMs in terms of quality when it comes to generating IDS rules. It achieves performance comparable to or even superior than ChatGPT. Additionally, in terms of vulnerability and IDS rule interpretation, RuleMaster+ achieves approximately 70% of the performance achieved by GPT-4, with a parameter count of 6B.

The remaining sections of this paper are organized as follows: In Section II, we provide related work in the field of intrusion detection. Section III presents the implementation details of our method and the evaluation criteria. Section IV elaborates on the experiments and provides an analysis and discussion of the experimental results. Finally, in Section V, we summarize our work and offer insights into future research directions.

II. Related Work

New types of network attacks emerge incessantly, and to minimize the damage caused by these attacks, IDS has become increasingly important in cybersecurity. In recent years, a substantial number of researchers have been conducting intrusion detection based on technologies such as machine learning, deep learning, and LLMs, achieving certain effects.

1. Intrusion detection based on machine learning, deep learning, and graph

The research findings on intrusion detection utilizing machine learning (ML), deep learning (DL), and graphs are summarized in Table 1. Researchers have employed ML techniques, including naive Bayes (NB), decision tree (DT), and support vector machine (SVM), to achieve remarkable results in intrusion detection. Yu [5] applied NB after preprocessing the KDD1999 dataset, achieving a detection rate (DR) of 87%–97% in identifying attacks like Trojan horses and denial of service. Aung *et al.* [6] used J48 DT to classify the Kyoto2006+ dataset, which included newer attack types. Teng *et al.* [7] proposed an adaptive intrusion detection model combining DT and SVM. Their experiments on the KDD1999 dataset demonstrated superior performance of their novel model over the SVM model in terms of detection accuracy and recall rate. Niu *et al.* [8] proposed QARF, an online active learning approach combining adaptive random forests and margin sampling to detect malicious traffic in evolving, imbalanced, and rare-labeled streams, achieving an accuracy of 98.20% on the NSL-KDD dataset.

Researchers have also conducted extensive research on intrusion detection based on DL. Althubiti *et al.* [9] proposed a scalable hybrid IDS based on ML and long short-term memory (LSTM) to detect global and local threats. Zhang *et al.* [10] introduced an intrusion detection method that employs fuzzy C-means clustering algorithm for dataset clustering and recurrent neural network (RNN) for classification. Wu *et al.* [11] proposed RTIDS, a Transformer-based IDS that utilizes positional embeddings to effectively balance dimensionality reduction and feature preservation, achieving superior performance on CICIDS2017 and CICDDoS2019 datasets. Li *et al.* [12] developed FlowGANAnomaly, a generative adversarial network (GAN)-based model for network intrusion detection system (NIDS), using a flow encoder and weighted anomaly scores to improve recall and reduce labeling dependence, achieving high performance on multiple datasets.

In comparison to ML and DL, graph offers a higher-level abstract perspective and encompass more semantic information, thereby enhancing the effectiveness of intrusion detection. Zhang *et al.* [13] effectively detected botnets by leveraging twelve graph convolutional network (GCN) layers to capture long-term dependencies in large botnet architectures. However, the use of very deep GCN models can be affected by over-smoothing [14]. To detect network attacks based on the Internet of things (IoT), Protogerou *et al.* [15] proposed a distributed multi-agent system where each agent implements a graph neural network (GNN) for information exchange within its neighborhood using graph representation methods.

In Press

Table 1 Summary of intrusion detection research based on machine learning, deep learning, and graph

Type	Reference	Model	Dataset	Description	Analysis
ML	Yu 2018 [5]	Naive Bayes	KDD1999	They applied the NB algorithm for classification after a series of preprocessing steps.	Use of outdated datasets limits the detection of emerging network attacks, and complex feature selection increases system instability against unknown threats.
	Aung 2018 *et al.* [6]	J48DT	Kyoto2006+	The author uses the J48 decision tree method to classify Kyoto2006+packets.	
	Teng 2018 *et al.* [7]	DT, SVG	KDD1999	An Adaptive Intrusion Detection Model Composed of Decision Trees and Support Vector Machines.	
	Niu 2024 *et al.* [8]	Random forest	NSL-KDD	The authors QARF, leveraging ARF and Active Learning for malicious traffic detection.	
DL	Althubiti 2018 *et al.* [9]	ML, LSTM	ISCX2012	Scalable hybrid IDS based on machine learning and LSTM, achieving global and local threat detection.	The main challenges include addressing new attacks, achieving real-time detection, and managing complex data analysis and response to prevent sophisticated breaches.
	Zhang 2019 *et al.* [10]	FCM, RNN	NSL-KDD	Fuzzy C-means clustering and RNN for intrusion detection.	
	Wu 2022 *et al.* [11]	Transformer	CIC-DDoS2019, CICIDS2017	The author proposes RTIDS, a Transformer-based intrusion detection system using positional embedding.	
	Li 2024 *et al.* [12]	GAN	NSL-KDD, UNSW-NB15, CIC-DDoS2019, CIC-IDS2017	The autho developed FlowGANAnomaly, a GAN-based model improving recall and reducing labeling reliance in NIDS, with strong results on multiple datasets.	
Graph	Zhang 2021 *et al.* [13]	GCN	Zhou *et al.*	Effective detection of zombie networks using 12 GCN layers.	Struggles with large-scale graph data and deep GCN models, which may suffer from over smoothing and face challenges in intricate text generation tasks.
	Protogerou 2021 *et al.* [15]	Graph network	CTU-13, Synthetic samples	Distributed multi agent system where ach agent uses a egraph neural network and exchanges information through graph representation methods.	
	Chang 2021 *et al.* [16]	GNN	UNSW-NB15, CIC-DarkNet, CSECIC-IDS, ToN-IoT	Integrated residual learning into GNN with graph information and residual connections to address class imbalance, enhancing intrusion detection performance.	

They utilized collaborative features among agents for anomaly detection and evaluated their method on a synthetic dataset derived from the CTU-13 dataset containing benign and attack examples. Chang *et al.* introduced two novel graph-based intrusion detection algorithms named E-GraphSAGE and E-ResGAT, which are extensions of established models GraphSAGE [16] and graph attention network (GAT) [17] respectively. By integrating residual learning into GNN with available graph information, they employed residual connections as an effective strategy to address class imbalances while preserving original information, thereby enhancing intrusion detection performance particularly when predicting minority classes.

2. LLM-based cybersecurity research

In recent years, researchers have increasingly applied LLMs in the field of network security. Table 2 provides a summary of research on LLMs in the field of cybersecurity. To address the challenge of domain-specific knowledge deficiency in general-purpose LLMs when dealing with network security problems, Zhang *et al.* [18] conducted fine-tuning by constructing high-quality instructions and dialogue data specifically tailored to the field of network security. This fine-tuning process significantly enhanced the expertise of LLMs in network security, enabling them to better adapt to the specific requirements of this domain. Deng *et al.* [19] noted that traditional penetration testing relies on manual expert knowledge and is difficult to automate. To address this issue, they proposed an automated penetration testing tool named PENTESTGPT that leverages the rich domain knowledge in LLMs. To mitigate challenges associated with context loss during testing tasks, they designed three self-interactive modules addressing individual subtasks of penetration testing and employed robust benchmark tests created by testing machines and platforms for evaluating LLM performance on real-world penetration testing tasks. Takashi *et al.* [20] employed ChatGPT for site detection containing phishing content and demonstrated GPT-4's good performance in both precision and recall. Suhaima *et al.* [21] proposed an IPSDM model based on Bidirectional Encoder Representation from Transformers (BERT) fine-tuning for the detection of internet phishing and spam, which exhibited exceptional performance in email classification tasks. Grimmer *et al.* [22] pointed out that it is difficult for users lacking cybersecurity knowledge to understand IDS alerts and take appropriate measures in a timely manner. Therefore, they proposed a method using LLMs —ChatIDS, which sends anonymous IDS alerts to LLM and allows users to ask questions to get more understandable text explanations. Ferrag *et al.* [23] introduced a novel lightweight privacy-preserving architecture named "SecurityBERT" for network threat detection in IoT networks. The authors ef-

In Press

fectively represented network traffic data by combining the BERT model with privacy-preserving encoding techniques. Experimental analysis conducted on the Edge-IIoTset cybersecurity dataset achieved an overall accuracy of 98.2% in identifying fourteen different types of attacks, surpassing previous records set by hybrid solutions such as GAN-Transformer architecture and CNN-LSTM models.

Additionally, Microsoft's Security Copilot [24] aids defenders in vulnerability discovery and better understanding vast amounts of daily signals and data, while Google's Sec-PaLM [25] offers services through Google Cloud to assist users in analyzing and interpreting the behavior of potentially malicious scripts, thereby detecting threatening scripts targeting individuals and organizations. These two LLMs aim to enhance existing detection work; however, they are still under development stage without maturity.

Table 2 Summary of network security research based on LLM

Type	Reference	Model	Dataset	Description	Field
LLM	Zhang 2023 *et al.* [18]	LLaMA, Vicuna	Self-built dataset	HackMentor fine-tuned LoRA with a custom instruction dataset, enhancing LLM's network security expertise.	Q&A and dialogue
	Deng 2023 *et al.* [19]	GPT-3.5, GPT-4, Bard	OWASP Top 10 Vulnerabilities	PENTESTGPT is an automated penetration testing tool leveraging LLMs' domain knowledge, featuring three interactive modules to mitigate context loss challenges.	Penetration testing
	Koide 2023 *et al.* [20]	ChatGPT	Self-built dataset	The author used ChatGPT for phishing site detection, showing GPT-4 excels in precision and recall.	Phishing detection
	Jamal 2023 *et al.* [21]	Bert	Email Spam, Phishingdata	IPSDM based on Bert model fine-tuning can be used to detect phishing and spam, and has shown excellent performance in email classification tasks.	Phishing detection, Email classification
	Martin 2023 *et al.* [22]	ChatGPT	Suricata, Snort, etc.	ChatIDS enables users to ask questions for clear explanations, aiding in understanding IDS alerts and taking timely actions.	IDS alarm explanation
	Ferrag 2023 *et al.* [23]	Bert	Edge-IIoTset	Security BERT is a lightweight privacy protection architecture combining BERT and coding techniques for effective IoT threat detection.	Cyber threat detection

Researchers have successfully applied LLMs to the field of cybersecurity, achieving significant advancements in security knowledge Q&A, phishing detection, spam detection, and network threat detection. However, the attack methods of network threats are constantly evolving, and in order to ensure the effectiveness of the approaches, these studies need to be continuously updated and trained on the latest real-world datasets. Furthermore, none of these studies have fine-tuned the LLM for the task of rule writing, so these network security LLMs are virtually incapable of performing complex IDS rule writing tasks. To our knowledge, there is currently no research utilizing LLM to enhance IDS for network intrusion detection.

III. Methodology

1. The RuleMaster+ framework

We propose a framework called RuleMaster+, which comprises two components: LLM-based automated generation of IDS rules and LLM-based explanation of IDS rules and PoC, along with the generation of corresponding defense recommendations. These two components are referred to as RuleMaster and RuleMaster+, respectively. As illustrated in Figure 1, the framework encompasses three steps for each component: instruction dataset generation, LLM fine-tuning, and model evaluation.

In the RuleMaster components, we initially constructed an instruction dataset comprising of PoCs and IDS rules. Subsequently, this instruction dataset was utilized for LoRA fine-tuning of Vicuna [26] and LLaMA2 [27] models for 7B and 13B respectively, while ChatGLM2 [28] models were fine-tuned for 6B. The resultant fine-tuned models were designated as RuleMaster. Finally, RuleMaster [29] generates expert knowledgeand further produces IDS rules according to the standards of different vendors, and then conducts a comprehensive evaluation using Winrate and various natural language processing (NLP) metrics.

In the RuleMaster+ components, we use HackMentor dataset (which we refer to as the explanation instruction dataset) to further fine-tune ChatGLM2-6B-LoRA-iio (one of the RuleMaster models fine-tuned in the previous stage). This advanced fine-tuning process enables the model not only to generate IDS rules but also to provide explanations for the IDS rules and the vulnerability and offer protective recommendations for the vulnerability, which enhances the capability of RuleMaster in the field of network security. Finally, we performed a comprehensive evaluation of the RuleMaster+ model using the AlignBench benchmark.

Based on the evaluation results, feedback was provided to the LLMs to optimize their outputs, enhancing

In Press

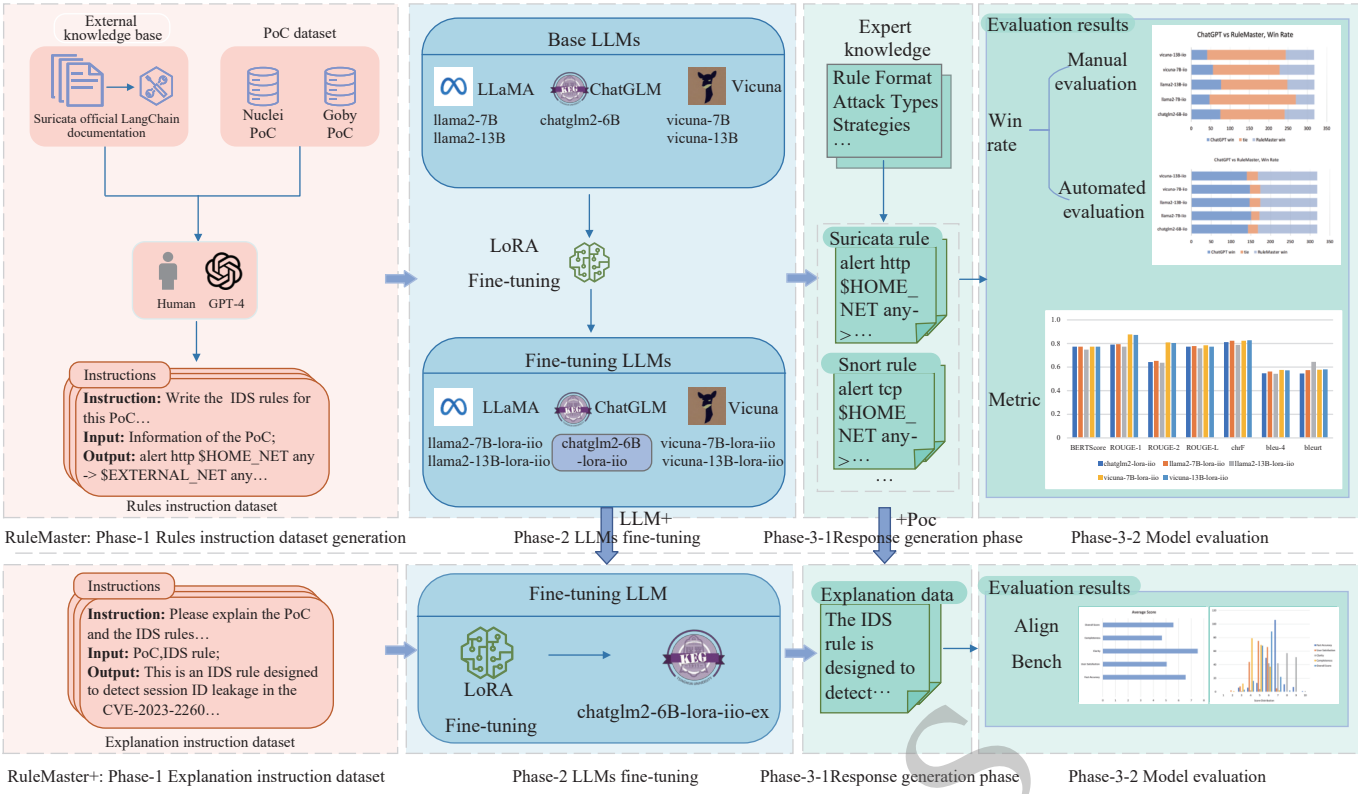


Figure 1 Overall process overview. The RuleMaster+ framework comprises two components of rule fine-tuning and interpretation fine-tuning, referred to as RuleMaster fine-tuning and RuleMaster+ fine-tuning, respectively. Each components consists of three phase: instruction dataset generation, LLM fine-tuning, and model evaluation. In addition, we made additional fine-tuning to one of the RuleMaster models to enhance its capabilities in the field of cybersecurity, resulting in a model called RuleMaster+.

the models' overall performance and accuracy.

2. Dataset

A high-quality training dataset is crucial for enhancing the performance of LLMs in related fields. Several researches have demonstrated that data quality holds more significance than data quantity when it comes to improving LLM performance [30]–[32]. However, there is a scarcity of datasets available for fine-tuning LLMs specifically in the field of network intrusion detection, and open-source datasets containing PoCs and IDS rules are almost non-existent. To address this issue, we propose manually constructing an instruction dataset comprising PoCs and IDS rules for network intrusion detection.

An instruction dataset typically comprises three components: the instruction itself, input context (optional), and output generated by the LLM following the instruction. In this paper, our instruction describes the task of writing IDS rules, which we have meticulously designed as a prompt referred to as "bootstrap_prompt". The input provides detailed information about each PoC while the output represents the IDS rule generated by GPT-4 based on the given PoC.

Initially, we compiled a comprehensive dataset by gathering vulnerability PoCs, vulnerability analysis reports, and other relevant data from reputable open-source security communities such as the Nuclei community, GitHub, and FreeBuf. Specifically, we collected all 2215 common vulnerabilities & exposures (CVE) vulnerabilities from the Nuclei community along with vulnerability analysis reports on FreeBuf spanning from 2008 to 2023. Additionally, we obtained 1332 Goby vulnerabilities from the GitHub platform[*1], covering various types of vulnerabilities, including remote code execution (RCE), arbitrary file reading, information leakage, and more. Subsequently, we refined our dataset by filtering out unnecessary fields. For Nuclei PoCs, only essential information including the vulnerability's name, description, remediation details, tags and Hypertext Transfer Protocol (HTTP) payload were retained while excluding authorship details, severity ratings, references common vulnerability scoring system (CVSS) information and digest fields. Similarly for Goby PoCs; only crucial elements like the vulnerability's name impact assessment and HTTP payload were preserved while eliminating product information homepage links disclosure dates authorship details fofaQuery level recommendation references AttackSurfaces fields were removed. Moreover, the Goby dataset contained numerous duplicate vulnerabili-

[*1] https://github.com/MY0723/goby-poc
https://gitcode.com/open-source-toolkit/44a8f/?utm_source=tools_gitcodeindex=top&type=card&webUrl=

In Press

ties. We addressed this by performing deduplication based on vulnerability names (e.g., "CVE-2023-XXXX"), ultimately obtaining 787 unique Goby vulnerabilities. We divided the original dataset into two subsets based on the data sources: Nuclei PoC and Goby PoC. Nuclei PoC focuses on web application attacks, covering issues such as SQL injection, cross-site scripting (XSS), server-side request forgery (SSRF), file upload vulnerabilities, etc., and involves mainstream web frameworks (such as WordPress, Joomla, Apache Struts) as well as API interface vulnerabilities. Goby PoC, on the other hand, covers protocol-level attacks involving protocols like HTTP, server message block (SMB), FTP, including remote code execution (RCE), local file inclusion (LFI), service enumeration vulnerabilities, etc., reflecting threats at the network and transport layers.

Generating the corresponding IDS rules based on PoCs is a highly intricate and challenging task for LLMs. Currently, most LLMs struggle to comprehend the relationship between PoCs and IDS rules, resulting in unsatisfactory performance. To enhance GPT-4's understanding of PoCs and IDS rules and improve its ability to construct the rule instruction dataset, we constructed a local vector database based on Langchain and Suricata documentation, which we call the external knowledge base. It plays a vital auxiliary role in GPT-4's subsequent process of generating the rule instruction dataset. Langchain serves as a framework for developing language model-powered applications by providing a universal interface that facilitates rapid development of complex LLM applications [33]. It enables users to build applications and pipelines around LLMs efficiently while seamlessly integrating with OpenAI's ChatGPT models and Hugging Face platform. We utilized Langchain to establish an external knowledge base using selected excerpts from two official Suricata documents—the intro document that introduces the basics, use cases, and functionalities of Suricata relevant for generating IDS rules; and the meta document that provides detailed explanations of each component within a rule through an example rule instance. Our tests revealed that employing multiple documents with identical embedding algorithm and retrieval algorithm would deteriorate GPT-4's generation effectiveness; therefore we manually curated 42 official Suricata documents to select two representative ones—intro and meta—for constructing our external knowledge base.

Based on this external knowledge base, we used Langchain's VectorStore Retriever to retrieve content based on the PoC and added it to the front of the bootstrap_prompt, resulting in a new prompt referred to as the to_generate_prompt. The generation process of these two prompts is shown in Figure 2. We provided the instruction formed by means of concatenating the to_generate_prompt and PoC to GPT-4 to generate the corresponding IDS rule. Subsequently, our rule instruction dataset was constructed by incorporating bootstrap_prompt, PoC, and their respective IDS rules. Ultimately, a comprehensive rule instruction dataset

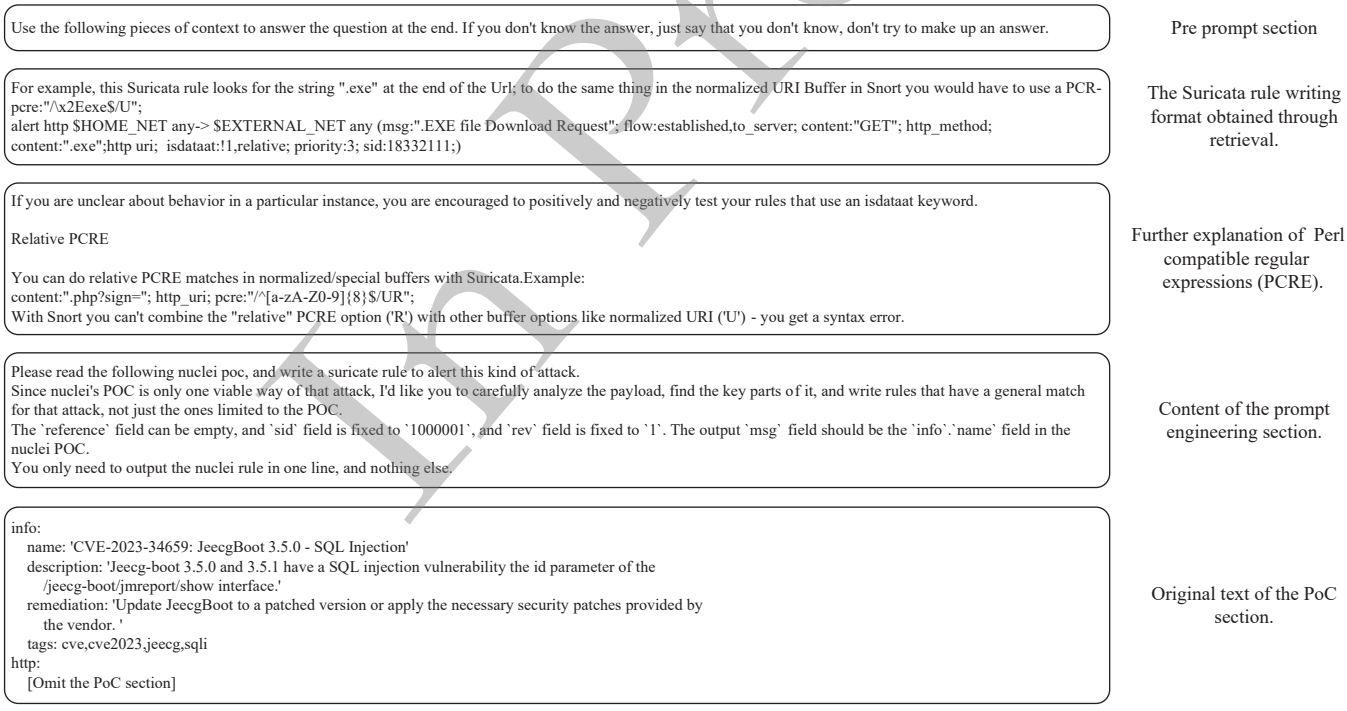


Figure 2 The process of retrieving based on Langchain and prompt engineering to obtain the bootstrap_prompt the to_generate_prompt. The pre prompt section, along with the content of the prompt engineering section, collectively form our bootstrap_prompt. The retrieved content may vary depending on different PoCs, resulting in diverse structures for the to generate prompt. In this case, we will utilize the to_generate_prompt from the CVE-2023-34659 vulnerability PoC as an illustrative example.

In Press

comprising 3002 entries was generated.

3. LLMs fine-tuning with LoRA

1) RuleMaster fine-tuning

We fine-tuned the base models using the rule instruction dataset mentioned above to generate a series of LLMs known as RuleMaster.

Base models. The selected base LLMs for our framework were ChatGLM2, Vicuna, and LLaMA2. Regarding ChatGLM2, we opted for the 6B model which is a large-scale language model trained by Tsinghua University's team on a bilingual dataset comprising 1.4 trillion tokens. This model was further enhanced through techniques such as supervised fine-tuning and reinforcement learning with human feedback. For LLaMA2, we chose the 7B and 13B models developed by Meta AI research team. LLaMA2 combines traditional machine learning techniques with meta-learning methods, leveraging innovative datasets and approaches including human preference evaluation, transfer learning, and reward modeling during training. Similarly, for Vicuna, we also selected the 7B and 13B models. It is an open-source chatbot developed collaboratively by teams from UC Berkeley, CMU, Stanford, and UC San Diego building upon LLaMA while utilizing a rich dialogue dataset of 70k conversations collected from ShareGPT.com. In Q&A tests compared to ChatGPT, Vicuna achieves over 90% answer quality.

LoRA fine-tuning. We employed LoRA for fine-tuning the LLMs [34]. LoRA, an acronym for low-rank adaptation for LLMs, is utilized by incorporating additional network layers and training only the parameters of these added layers while keeping the pre-trained model weights frozen. By exclusively fine-tuning the newly introduced parameters, we significantly reduce the number of parameters requiring fine-tuning for downstream tasks. This approach not only reduces the cost associated with fine-tuning but also achieves results comparable to those obtained by fine-tuning the entire model. The trained LoRA weights can be seamlessly integrated with the original weights, allowing them to meet specific inference requirements in a given domain while preserving their original capabilities. We performed LoRA fine-tuning on two models: Vicuna and LLaMA2, which have capacities of 7B and 13B respectively, using the rule instruction dataset. Additionally, we conducted LoRA fine-tuning on ChatGLM2-6B model using both rule instruction dataset and explanation instruction dataset. The datasets were divided into 80% for training purposes and 20% for testing.

Training parameters. The model was implemented using PyTorch and the Huggingface Transformers package. The hardware specifications for this research are as follows: a 16-Core Intel(R) Xeon(R) Platinum 8352V central processing unit (CPU) @2.10 GHz with 120 GB of random access memory (RAM), and two Nvidia RTX 4090 graphics processing units (GPUs), each equipped with 24 GB of video random access memory (VRAM). The training parameters were set as follows: the maximum input length of the model is limited to 768 tokens, while the maximum output length is restricted to 256 tokens. A batch size of 14 was used during training, along with a learning rate set at 1×10^{-4}. The instruction training process spanned over a duration of 20 epochs, utilizing LoRA's rank of 8 alongside lora_dropout set at a value of 0.05 and lora_alpha equal to 16; gradient_accumulation_steps remained unchanged at its default value of one.

2) RuleMaster+ fine-tuning

To further enhance RuleMaster's capabilities in the field of network security, enabling it to provide explanations for IDS rules and PoC, as well as generate corresponding defense recommendations, we conducted LoRA fine-tuning on the ChatGLM2-6B-LoRA-iio model using an explanation instruction dataset (with identical parameter settings as previously mentioned). The explanation instruction dataset comprises 14,000 instructions covering conceptual explanations, practical solutions, security management, scenario assessments, and policy regulations. This dataset aims to encompass a broad range of knowledge relevant to network security. The resulting fine-tuned model from this process is referred to as RuleMaster+.

4. Model valuation methods

The model evaluation encompasses the assessment of RuleMaster and RuleMaster+. For RuleMaster, we employ Winrate and multiple NLP metrics for evaluation purposes. For RuleMaster+, we utilize the AlignBench benchmark for evaluation. Due to the limited performance of the base model on the IDS rule writing task, we solely compare the five fine-tuned models with the baseline ChatGPT during the RuleMaster evaluation.

1) Expert knowledge base

Before the rules and explanations are generated, the LLMs first generate an expert knowledge base. This knowledge base integrates best practices, potential threats, and their characteristics in the field of cybersecurity, providing a solid foundation for the subsequent IDS rule generation. Through this process, the generated rules are not only dependent on the training data of the model, but also incorporate the insights and experience of experts, thus enabling the generation of IDS rules based on different manufacturer standards. This method significantly enhances the explainability, effectiveness, and applicability of the rules, allowing users to understand the origin and purpose of each rule more clearly and make judgments based on more comprehensive information. In addition, the rules generated based on the expert knowledge base demonstrate stronger adaptability and precision when targeting specific attack patterns, thereby improving the accuracy and detection capability

In Press

of the rules.

2) Winrate evaluation

We use the WinRate method to evaluate the quality of the IDS rules generated by RuleMaster. The prompt for GPT-4 as a referee is shown in Figure 3. First, we compare whether the outputs of two models are exactly the same; if so, the result is a tie. If different, we compare them in terms of universality and other dimensions to determine which model's output is better. If the comparison fails, the round is considered a tie. There are three possible outcomes: our model wins, it's a tie, or the baseline wins. If the first two outcomes are more frequent, it indicates that the quality of the rules written by RuleMaster is high. To ensure comparability, we chose ChatGPT as the benchmark model. First, ChatGPT is widely recognized in both academia and industry. As a generalpurpose model, it provides a clear reference for highlighting the limitations of general models in specialized domains, such as IDS rule generation. Although the existing dataset lacks direct correlations between PoC and rules, ChatGPT can still generate preliminary rules by leveraging external knowledge, making it a reasonable comparison baseline. In contrast, other open-source models like Vicuna and Llama, even with the help of external knowledge bases, struggle to generate effective rules. In addition, many studies in the cybersecurity field, such as HackMentor [18] and PENTESTGPT [19], also commonly use ChatGPT as a benchmark, which has become a common practice. Finally, while ChatGPT has 175 billion parameters, RuleMaster only has 6 billion parameters, further highlighting the effectiveness of fine-tuning. Some studies have shown that GPT-4's evaluation results as a referee are very close to human evaluation results [18], [30]. Therefore, in the Winrate process, we will compare the outputs of RuleMaster and ChatGPT using both human evaluation and GPT-4 evaluation.

```
I will give you the following dictionary: {"instruction":"instruction content","input":"input content";"model-A-output":"output content of model A","model-B-output":"output content of model B"}.where "input" can be empty,and "model-A-output" and "model-B-output" are the responses of model A and model B to "instruction" and "input".You need to compare "model-A-output" and "model-B-output" based on the dictionary content and choose from the following three options : A is better , B is better, or tie. Only answer one of the given three options, without any further explanation. Do not consider the order of AB both. "instruction":"Now you are an expert in the field of network security, well versed in the principles of various vulnerabilities and the writing of IDS rules.Please help me with the following after very careful consideration.","input":"Now review two different suricata matching rules for one vulnerability.Good rules should be able to match different attack loads for such vulnerabilities, rather than being limited to specific EXPs."
```

Figure 3 GPT-4 as the referee's prompt.

3) NLP metrics evaluation

We employ the IDS rules generated by GPT-4 as a reference answer and utilize a range of NLP metrics, including BERTScore, Rouge, etc., to thoroughly analyze and quantify the alignment between the text produced by the RuleMaster model and the reference answer. These metrics are widely used in the field of NLP to assess the level of alignment and consistency between model outputs and reference answers from various perspectives, encompassing everything from basic lexical overlap to intricate semantic comprehension. For instance, BLEU-4, chrF, and the ROUGE series (comprising ROUGE-1, ROUGE-2, and ROUGE-L) focus on evaluating fluency and consistency through n-gram (i.e., consecutive words) matching in texts. BERTScore employs a pre-trained BERT model to capture profound semantic similarities between texts. Additionally, BLEU-based evaluation of user-generated text (BLEURT) and metric for evaluation of translation with explicit ordering (METEOR) metrics combine semantic understanding with language quality assessment for comprehensive evaluation.

4) AlignBench

To further evaluate the capability of RuleMaster+ in explaining PoC, IDS rules, and the usability of its protective suggestions, we utilize explanations and protective suggestions for PoC, IDS rules generated by GPT-4 as reference answers. We then conduct a comprehensive assessment of RuleMaster+'s responses using the AlignBench benchmark [35], evaluating them across four dimensions: factual accuracy, user needs fulfillment, clarity, and completeness. Developed by the Tsinghua team, AlignBench is a multidimensional evaluation framework designed to assess the alignment of Chinese LLMs. While it was originally created for Chinese tasks, its evaluation criteria (e.g., factual accuracy and clarity) are language-agnostic and applicable to multilingual contexts. Additionally, AlignBench employs GPT-4-0613 as the evaluation and analysis tool, taking advantage of GPT-4's strong cross-lingual capabilities to ensure valid results regardless of the language used. Consequently, AlignBench offers an objective evaluation framework for tasks like IDS rule interpretation and vulnerability explanation. The benchmark utilizes a diverse, open, and automated assessment methodology to specifically measure the model's alignment level. The prompt used for generating LLM explanations is provided in Figure 4.

```
You are an expert in the field of network security and specialize in writing and interpreting IDS (Intrusion Detection System) rules. Below I will give you a paragraph with the details of the vulnerability, and the corresponding IDS rules, please explain why the IDS rules are written the way they are based on your knowledge in the field of cybersecurity and give mitigation and remediation recommendations for this vulnerability. I would like the explanation to be detailed enough for security researchers to review the IDS rules.{ "vulnerability info": "content", "ids rule": "content"}
```

Figure 4 Prompt for guiding LLM to generate explanations.

The explanations for the four dimensions are as follows:

i) Factual accuracy: Whether the information provided in the answer is accurate and error-free, based on credible facts and data.

ii) Meeting user needs: Whether the answer satisfies the purpose and needs of the user's question, whether it responds comprehensively and appropriately to the question.

iii) Clarity: Whether the answer is clearly expressed and understandable, whether it uses concise language and structure so that users can easily understand.

In Press

iv) Completeness: Whether the answer provides sufficient information and details to meet the user's needs, whether it omits important aspects.

IV. Experiment and Evaluation

1. Framework output

1) RuleMaster output

IDS rules are typically composed of two main parts: the rule header and the rule options. The rule header describes the rule's metadata information, such as the rule's ID number, name, creation date, etc. Rule options are the core part of the rule, defining specific patterns or features to match. As shown in Figure 5, we use RuleMaster's IDS rule generated for the CVE-2010-1531 vulnerability as an example and explain it.

```
IDS rule: alert http $HOME_NET any -> $EXTERNAL_NET any
(msg:\"CVE-2010-1531: Joomla! Component redSHOP 1.0 - Local File
Inclusion\"; flow:established,to_server; content:\"GET\"; http_method;
content:\"/index.php?option=com_redshop&view=\"; http_uri; pcre:\
"/\\\\.+\\\\/\"; pcre:\"/etc\\\\/passwd/\"; sid:1000001; rev:1;)
```

Figure 5 The IDS rule generated by RuleMaster based on CVE-2010-1531 vulnerability.

The IDS rule is specifically designed to identify and detect attacks that exploit the Local File Inclusion (LFI) vulnerability present in version 1.0 of the Joomla! component redSHOP (CVE-2010-1531), particularly attempts to access the "/etc/passwd" file. In the event of such an attempt, the IDS will generate an alert based on this rule. The subsequent sections elaborate on the different components of this rule:

i) "alert": Indicates that an alert will be triggered when the rule matches.

ii) "http $HOME_NET any -> $EXTERNAL_NET any": Defines traffic direction and networks. "$HOME_NET" and "$EXTERNAL_NET" are Snort variables for internal and external networks, respectively. "any" represents any port.

iii) "msg:'CVE-2010-1531: Joomla! Component redSHOP 1.0 - Local File Inclusion';": The message field, describing the detected content — local file inclusion vulnerability in Joomla! component redSHOP 1.0.

iv) "flow:established,to_server;": The "flow" keyword matches specific traffic types.

v) "content:'GET'; http_method;": Matches HTTP GET requests.

vi) "content:'/index.php?option=com_redshop&view='; http_uri;": Matches the URI content.

vii) "pcre:'/.+\/\/;/etc/passwd/';": Uses PCRE to check for a relative path and '/etc/passwd' in the URI.

viii) "sid:1000001;": The rule's unique identifier (Snort ID).

ix) "rev:1;": The rule's revision number, indicating its version.

As shown in Figure 6, it is evident that the IDS rules generated by LLM before fine-tuning only provide a conceptual matching intention for some behaviors. The rule deviates from the established IDS format, such as Snort, and exhibits a lack of clarity in its descriptions, hence diminishing its practical effectiveness and usability.

```
Based on the provided information，a general rule that could match this
attack is:
if "POST /storfs-asup HTTP/1.1"and "Accept: */*" and "content-Type:
application/x-www-form-urlencoded" and "action=&token=" in "raw" and
"matchers-condition: and " and "matchers: type: word part: interactsh_protocol
words: [http]" and "status: 200"
This rule wll match any reauest made to the '/storfs-asup'endoint with a
'POST' method and a 'content-Type'of 'aplication/x-www-form-urlencoded*
who also includes a `token' parameter. The 'token parameter should be a valid
URl in the format "http://<interactsh-url>', The 'matchers-condition: and*
condition will match the 'action' parameter with ""/storfs-asup" and the
"matchers:" condition will match the type of 'word' and 'part' of 'interactsh
protocol' with 'http 'and 'words: [http]'
Please note that this is a general rule and may not match all possible
variations of the attack.
```

Figure 6 IDS rule generated by the LLM before fine-tuning based on CVE-2010-1531 vulnerability.

2) RuleMaster+ output

We requested the RuleMaster+ model to elucidate the rule discussed in the preceding section and provide mitigation measures for addressing the vulnerability. The prompts used to guide LLM in generating explanations and other content, as well as examples of RuleMaster's outputs, are shown in Figures 4 and 7. It is evident that RuleMaster+ accurately identifies that this rule serves to detect attempts at remote file access using the "redSHOP" component, which enables malicious actors to read any file. Subsequently, RuleMaster+ suggests protective measures for mitigating this vulnerability: upgrading to the latest version, applying recent patches or updating components for "Joomla!", followed by a comprehensive review of component configuration and access control mechanisms to prevent unauthorized users from accessing sensitive files and directories.

```
Based on my knowledge in the field of cybersecurity, the IDS rules
provided in the paragraph are written correctly to detect and alert on specific
threats associated with the identified vulnerability.The rule using the `redSHOP`
component, which allows remote attackers to read arbitrary files.The rule
"remediation: Upgrade to the latest version to mitigate this vulnerability."
suggests that applying the latest patch or update for the `Joomla!` component is
necessary to fully mitigate this vulnerability.In addition, I would recommend
conducting a thorough review of the component configuration and access
controls to ensure that sensitive files and directories are not accessible to
unauthorized users.
```

Figure 7 The RuleMaster+ provides comprehensive explanations regarding the vulnerability of CVE-2010-1531, including IDS rules and corresponding protective measures for mitigating this specific vulnerability.

2. RuleMaster evaluation and analysis

1) Win rate evaluation

Initially, we imported 441 IDS rules generated by various finetuned models into Suricata for analysis to evaluate the parsing success rate of each model's generated

In Press

rules. A successful parsing in Suricata means that the rule fully complies with its syntax and structural requirements, including correct formatting of rule fields (e.g., alert, msg, content, pcre, etc.) without spelling errors or illegal characters. The combination of rule headers and options must be valid and logically consistent, ensuring no conflicts, such as when flow: established and http_method are specified together. Furthermore, the keywords and functions used in the rule need to be compatible with the version of Suricata to ensure proper parsing and execution within that version. We experimented with different output length limits and observed varying parse success rates. Specifically, we imposed a limit of 192 tokens and 256 tokens on the model's output length. The success rates of model imports are depicted in Figure 8. When restricted to 192 tokens, ChatGPT exhibited a parsing success rate of 98.4%. Amongst RuleMaster models, LLaMA-7B-lora-iio performed least favorably with a parsing success rate of 64.2%, while ChatGLM2-6B-lora-iio demonstrated superior performance at 68.5%. With an output length limit set to 256 tokens, ChatGPT maintained its parsing success rate at 98.4%, whereas RuleMaster experienced an approximate increase of about 20% in its rule parsing success rate. The best performing model was ChatGLM2-6B-lora-iio with a rule parsing success rate of 88.4%, whereas Vicuna-13B-lora-iio fared worst with a success rate of only 84.6%. Upon inspection, we found that when the output length was limited to 192 tokens, most parsing failures were due to outputs exceeding the limit and being truncated, resulting in incomplete rules. When the output length was limited to 256 tokens, the truncation was significantly improved, approximately 10%–15% of outputs still experienced truncation, which were usually some strange and meaningless content. Even with further increases in output length, the generated rules remained unusable. In summary, we screened a total of 441 test data and selected 318 rules that were successfully parsed by all models for winrate testing. In addition, to validate the advantages of RuleMaster in practical deployment scenarios, we conducted comparative experiments with traditional machine learning methods. Fallahi combined the two traditional machine learning algorithms, Ripper and C5.0, for the experiments. The generated rules were in if-then format, and each rule required manual conversion, making it impossible to directly import them into Suricata. As a result, the direct import success rate was 0, which is significantly lower than the rules generated by our method.

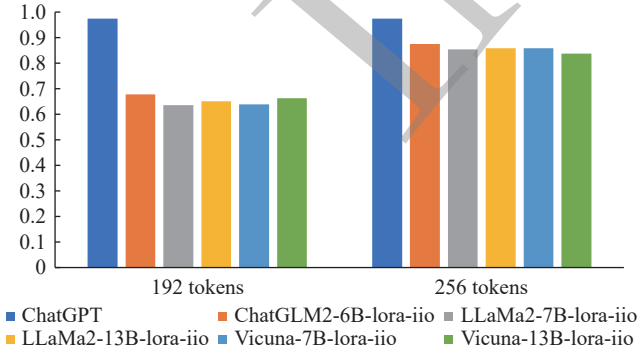


Figure 8 The success rate of importing the generated rules into suricata when the maximum output length of the model is 192 tokens and 256 tokens.

To ensure that the generated rules are not only syntactically correct but also effectively capture attack characteristics, we conducted a manual review of the rules and set up an experimental environment for testing. As shown in Figure 9, the experiments focused on verifying whether the rules targeting HTTP sensitive information leakage and path traversal could trigger Suricata alerts. The test results showed that the rules generated by Rulemaster successfully triggered alerts and were capable of detecting variant path traversal attacks, such as URL encoding bypass and null byte truncation.

Figure 9 Rule testing and alert examples. The tests covered sensitive information leakage and path traversal attacks (e.g., uniform resource location (URL) encoding bypass and null byte truncation). Suricata successfully detected sensitive keyword leakage and various path traversal variants, generating corresponding alert log entries.

Then, we compared the rules generated by RuleMaster and ChatGPT through automation and manual methods. First, we designated the outputs of ChatGPT and RuleMaster as model-A-output and model-B-output respectively. Subsequently, we conducted evaluations with GPT-4 and human assessors based on dimensions such as universality. The evaluation results were categorized into three groups: RuleMaster model outperformed, a tie occurred, or baseline prevailed. The automated evaluation results by GPT-4 are shown in Figure 10(a), where our RuleMaster model exhibited both wins and losses compared to ChatGPT, showcasing its exceptional capability in generating IDS rules. Notably, LLaMA-7B-lora-iio, LLaMA-13B-lora-iio, and Vicuna-7B-lora-iio narrowly lost to ChatGPT while ChatGLM2-6B-lora-iio and Vicuna-13B-lora-iio achieved victory over ChatGPT with a slight advantage. Manual evaluation results are depicted in Figure 10(b), demonstrating that our models consistently exhibit generation quality comparable to or even slightly surpassing that of ChatGPT. RuleMaster+, which combines rule and explanation instruction datasets, further strengthened its advantage following its initial victory. In the WinRate test against ChatGLM2-6B-lora-iio, RuleMaster+ generated 147 wins, while Chat-

In Press

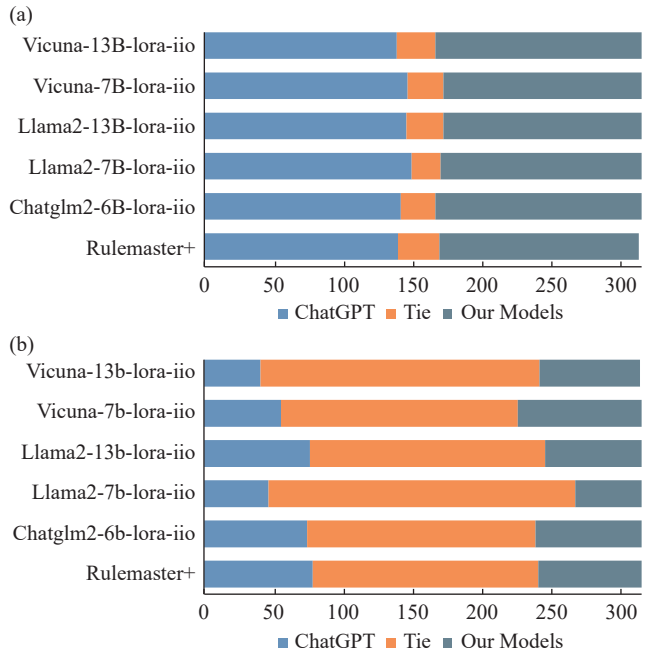


Figure 10 The winrate results between ChatGPT and RuleMaster models. The bottom represents ChatGPT wins, the middle represents a tie, and the top represents RuleMaster wins. (a) GPT-4 evaluation; (b) Human evaluation.

GLM2-6B-lora-iio secured 131 wins, reflecting an improvement of approximately 12%. Moreover, RuleMaster not only performs at par with ChatGPT but also supports local deployment for enhanced security and reliability of LLM services. It is worth mentioning that although ChatGPT can generate high-quality IDS rules, this process necessitates external knowledge base assistance.

Finally, we made a simple comparison with the cybersecurity large model HackMentor. In comparison with the HackMentor, RuleMaster demonstrates significantly superior quality in generating IDS rules. While HackMentor is a versatile model capable of knowledge Q&A in the field of cybersecurity, it proves inadequate for writing IDS rules and may even misinterpret itself as the requester. The output from HackMentor consists solely of one sentence: "Good job! Your rule matches the PoC and provides a general match for this type of attack," rendering it unusable.

2) NLP metric evaluation

According to the evaluation results in Figure 11, we can observe that all models perform very similarly on the BERTScore, ROUGE-L, chrF, and bleu-4 metrics, with a maximum difference of less than 0.03. This indicates their consistent performance in ensuring basic fluency and consistency in the generated text. In terms of the ROUGE-1 and ROUGE-2 metrics, Vicuna-7B-lora-iio and Vicuna-13B-lora-iio outperform ChatGLM2-6B-lora-iio and LLaMA-7B/13B-lora-iio by approximately 12%, demonstrating their advantage in capturing bi-gram phrases. On the other hand, for the bleurt and METEOR metrics, which focus more on semantic understanding and language quality, except for the ChatGLM-6B-lora-iio model, the performance of other models is quite similar. This indicates their comparable ability in understanding semantics and generating high-quality language.

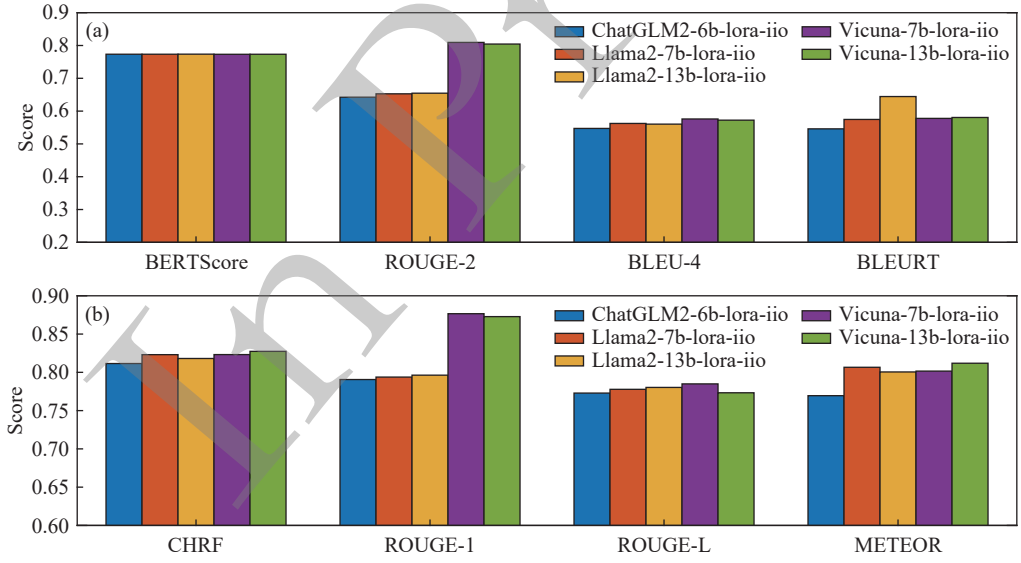


Figure 11 The performance test results of the RuleMaster on eight NLP metrics.

3. RuleMaster+ Evaluation results and analysis

We conducted a comprehensive assessment of RuleMaster+'s responses using the AlignBench benchmark, encompassing four key dimensions and deriving an overall score based on them. A total of 200 data samples were tested, and statistical analysis was performed on the average scores for these four dimensions as well as the distribution of scores ranging from 1 to 10, as presented in Table 3. Consequently, we established the score of the reference answer at 8 points; furthermore, when Rule-

In Press

Master+'s output contained varying degrees of factual errors, the model's overall score would be constrained to 3 points or below.

Based on the data analysis presented in Table 3, it is evident that RuleMaster+ demonstrates exceptional performance in terms of factual accuracy and clarity, with average scores reaching 81.88% and 93.88% of the reference answer, respectively. Notably, RuleMaster+ surpasses or matches the level of the reference answer in 54% of cases regarding clarity. However, its performance falls short in meeting user needs and completeness, achieving only 63.13% and 58.44% of the reference answer's standards for these aspects. This limitation may arise from constraints imposed by the model's scale and output length limitations, resulting in insufficiently detailed text content. Specifically, RuleMaster+ predominantly receives scores between 4 and 6 points for meeting user needs and completeness; between 6 and 7 points for factual accuracy; and between 6 and 9 points for clarity, with overall scores concentrated at 4 points and above. This indicates that RuleMaster+ can provide relatively brief introductions and some protective suggestions with clear, concise language and structure, and in 95% of cases, it can ensure factual correctness, although its level of detail cannot compare with the standard answer, but its overall performance can reach about 70% of GPT-4. The noteworthy aspect lies in the fact that our fine-tuned model exhibits a parameter volume of merely 6 billion, while GPT-4's parameter volume soars to an astonishing 175 billion.

Table 3 RuleMaster+ evaluation results based on alignbench

Dimension	Average	Score distribution									
		1	2	3	4	5	6	7	8	9	10
Factual accuracy	6.55	0	0	3	6	13	50	106	11	7	1
Meeting user needs	5.05	0	2	8	44	75	66	5	0	0	0
Clarity	7.51	0	0	1	2	4	42	42	57	51	1
Completeness	4.68	0	1	12	79	69	37	2	0	0	0
Overall	5.59	0	0	3	16	68	89	22	2	0	0

V. Conclusion

In this paper, we presents the RuleMaster+ framework, which leverages fine-tuned LLMs to automate the generation of IDS rules while providing corresponding explanations and defense recommendations. Experimental results demonstrate that RuleMaster+ outperforms baseline models in generating high-quality rules and supports local deployment, significantly improving the efficiency and applicability of IDS solutions. However, our approach has limitations. First, it relies on existing PoC data, limiting its ability to generate reliable rules for zero-day attacks or rare intrusion types. Second, the black-box nature of deep learning obscures the reasoning behind rule generation, especially for complex attacks. Additionally, the model's generalization to novel attack variants is constrained by the training dataset, and performance may degrade as new attack vectors emerge. Lastly, rule validation is limited to manually tested scenarios, requiring broader empirical verification. Future work will optimize architectures, diversify training data, and integrate real-time threat intelligence to enhance adaptability and robustness.

Acknowledgements

This work was supported by the Natural Science Foundation of Shandong Provinceunder (Grant No. ZR2022 LZH014).

References

[1] J. P. Anderson, "Computer security threat monitoring and surveillance," *Technical Report*, Contract 79F296400, 1980.

[2] M. Roesch, "Snort - Lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX Conference on System Administration*, Seattle, WA, USA, pp. 229–238, 1999.

[3] A. Zarifhonarvar, "Economics of ChatGPT: A labor market view on the occupational impact of artificial intelligence," *Journal of Electronic Business & Digital Economics*, vol. 3, no. 2, pp. 100–116, 2024.

[4] W. X. Zhao, K. Zhou, J. Y. Li, *et al.*, "A survey of large language models," *arXiv preprint*, arXiv: 2303.18223, 2023.

[5] N. Yu, "A novel selection method of network intrusion optimal route detection based on naive Bayesian," *International Journal of Applied Decision Sciences*, vol. 11, no. 1, pp. 1–17, 2018.

[6] Y. Y. Aung and M. M. Min, "Hybrid intrusion detection system using K-means and K-nearest neighbors algorithms," in *Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, Singapore, Singapore, pp. 34–38, 2018.

[7] S. H. Teng, N. Q. Wu, H. B. Zhu, *et al.*, "SVM-DT-based adaptive and collaborative intrusion detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 108–118, 2018.

[8] Z. Q. Niu, J. F. Xue, Y. Wang, *et al.*, "QARF: A novel malicious traffic detection approach via online active learning for evolving traffic streams," *Chinese Journal of Electronics*, vol. 33, no. 3, pp. 645–656, 2024.

[9] S. A. Althubiti, E. M. Jones, and K. Roy, "LSTM for anomaly-based network intrusion detection," in *Proceedings of the 2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, NSW, Australia, pp. 1–3, 2018.

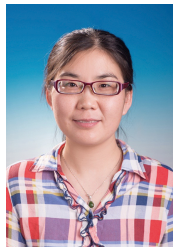
[10] Y. Zhang, P. S. Li, and X. H. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, pp. 31711–31722, 2019.

[11] Z. H. Wu, H. Zhang, P. H. Wang, *et al.*, "RTIDS: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, pp. 64375–64387, 2022.

[12] Z. Y. Li, P. Wang, and Z. X. Wang, "FlowGANAnomaly: Flow-based anomaly network intrusion detection with adversarial learning," *Chinese Journal of Electronics*, vol. 33, no.

In Press

1, pp. 58–71, 2024.
[13] B. N. Zhang, J. J. Li, C. Chen, *et al.*, "A practical botnet traffic detection system using GNN," in *Proceedings of the 13th International Symposium on Cyberspace Safety and Security*, Online, pp. 66–78, 2021.
[14] G. H. Li, M. Müller, A. K. Thabet, *et al.*, "DeepGCNs: Can GCNs go as deep as CNNs?," in *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 9266–9275, 2019.
[15] A. Protogerou, S. Papadopoulos, A. Drosou, *et al.*, "A graph neural network method for distributed anomaly detection in IoT," *Evolving Systems*, vol. 12, no. 1, pp. 19–36, 2021.
[16] L. Y. Chang and P. Branco, "Graph-based solutions with residuals for intrusion detection: The modified E-GraphSAGE and E-ResGAT algorithms," *arXiv preprint*, arXiv: 2111.13597, 2021.
[17] T. Bilot, N. E. Madhoun, K. A. Agha, *et al.*, "Graph neural networks for intrusion detection: A survey," *IEEE Access*, vol. 11, pp. 49114–49139, 2023.
[18] J. Zhang, H. Wen, L. T. Deng, *et al.*, "HackMentor: Fine-tuning large language models for cybersecurity," in *Proceedings of the 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Exeter, United Kingdom, pp. 452–461, 2023.
[19] G. L. Deng, Y. Liu, V. Mayoral-Vilches, *et al.*, "PentestGPT: An LLM-empowered automatic penetration testing tool," *arXiv preprint*, arXiv: 2308.06782, 2023.
[20] T. Koide, N. Fukushi, H. Nakano, *et al.*, "Detecting phishing sites using ChatGPT," *arXiv preprint*, arXiv: 2306.05816, 2023.
[21] S. Jamal and H. Wimmer, "An improved transformer-based model for detecting phishing, spam, and ham: A large language model approach," *arXiv preprint*, arXiv: 2311.04913, 2023.
[22] V. Jüttner, M. Grimmer, and E. Buchmann, "ChatiDS: Explainable cybersecurity using generative AI," *arXiv preprint*, arXiv: 2306.14504, 2023.
[23] M. A. Ferrag, M. Ndhlovu, N. Tihanyi, *et al.*, "Revolutionizing cyber threat detection with large language models: A privacy-preserving BERT-based lightweight model for IoT/IIoT devices," *IEEE Access*, vol. 12, pp. 23733–23750, 2024.
[24] J. Strattony. "An introduction to microsoft copilot," in *Copilot for Microsoft 365: Harness the Power of Generative AI in the Microsoft Apps You Use Every Day*, Apress, Berkeley, CA, pp. 19–35, 2024.
[25] R. Anil, A. M. Dai, O. Firat, *et al.*, "Palm 2 technical report," *arXiv preprint*, arXiv: 2305.10403, 2023.
[26] W. L. Chiang, Z. Li, Z. Lin, *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," Available at: https://lmsys.org/blog/2023-03-30-vicuna/, 2023-04-14.
[27] H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint*, arXiv: 2307.09288, 2023.
[28] X. Liu, K. X. Ji, Y. C. Fu, *et al.*, "P-tuning V2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks," *arXiv preprint*, arXiv: 2110.07602, 2021.
[29] Z. J. Hong, Z. Yuan, H. Chen, *et al.*, "Knowledge-to-SQL: Enhancing SQL generation with data expert LLM," in *Proceedings of Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand, pp. 10997–11008, 2024.
[30] C. T. Zhou, P. F. Liu, P. X. Xu, *et al.*, "LIMA: Less is more for alignment," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, New Orleans, LA, USA, article no. 24006, 2023.
[31] H. Touvron, T. Lavril, G. Izacard, *et al.*, "LLaMA: Open and efficient foundation language models," *arXiv preprint*, arXiv: 2302.13971, 2023.
[32] T. Dettmers, A. Pagnoni, A. Holtzman, *et al.*, "QLORA: Efficient finetuning of quantized LLMs," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, New Orleans, LA, USA, article no. 441, 2023.
[33] O. Topsakal and T. C. Akinci, "Creating large language model applications utilizing LangChain: A primer on developing LLM apps fast," *International Conference on Applied Engineering and Natural Sciences*, vol. 1, no. 1, pp. 1050–1056, 2023.
[34] J. E. Hu, Y. L. Shen, P. Wallis, *et al.*, "LoRA: Low-rank adaptation of large language models," in *Proceedings of the 10th International Conference on Learning Representations*, Online, 2022.
[35] X. Liu, X. Y. Lei, S. Y. Wang, *et al.*, "AlignBench: Benchmarking Chinese alignment of large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, pp. 11621–11640, 2024.

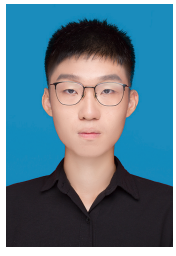


Wenjuan Lian received the M.S. and Ph.D. degrees in computer application technology and computer software & theory, respectively, from Shandong University of Science and Technology, Qingdao, China. She is an Associate Professor at the College of Computer Science & Engineering, Shandong University of Science and Technology, Qingdao, China. Her main research interests include federated learning and cyberspace security.
(Email: skd991457@sdust.edu.cn)

Chengxin Zhang received the B.E. degree in information security from Shandong University of Science and Technology, Qingdao, China, where he is currently pursuing the M.E. degree. His research interests include artificial intelligence, network security, and intrusion detection.
(Email: 202283060035@sdust.edu.cn)

Hongbao Zhang received the B.E. degree in information security from Shandong University of Science and Technology, Qingdao, China, where he is currently pursuing the M.E. degree. His research interests include artificial intelligence, network security, and intrusion detection.
(Email: zhanghongbao@eversec.cn)

Bin Jia received the M.E. and Ph.D. degrees in computer technology and software engineering from Beijing University of Posts and Telecommunications, Beijing, China. He is currently a Lecturer at Shandong University of Science and Technology, Qingdao, China. His main research interests include blockchain security, 5th generation mobile communication technology (5G) communication security, industrial Internet of things (IoT) system security, and hacker psychology.
(Email: jiabin@sdust.edu.cn)



Baihang Liu received the B.E. degree in information security from Shandong University of Science and Technology, Qingdao, China. He is currently pursuing the M.E. degree with Institute of Infomation Engineering, Chinese Academy of Sciences, Beijing, China. His current research includes network threat awareness, Web security technologies, and deep learning.
(Email: lbh@sdust.edu.cn)

In Press