Vol. 14 (2026): Proceedings of the 2nd International Conference on Artificial Intelligence, Modern Engineering and Environmental Sustainability (IC-AIMEES 2026)
E-ISSN: 3105-0522 P-ISSN：3105-0514
DOI: https://doi.org/10.70267/ ic-aimees.2026021523

A New Paradigm of Active Defense System Based on Large Language Models

Shujing Xiang*

College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430074, Hubei, China

**Corresponding author: Shujing Xiang*

Abstract

With the continuous evolution of cyber attack technologies and the persistent expansion of attack scales, traditional passive defense systems are confronting unprecedented challenges. As a pivotal breakthrough in the field of artificial intelligence, Large Language Models (LLMs) offer a novel technical pathway for the construction of active defense systems. This paper systematically reviews LLM technologies tailored to active defense systems. Firstly, it briefly delineates the fundamental principles of LLMs. Secondly, it conducts an in-depth exploration of the core technical mechanisms of LLMs and focuses on analyzing their unique advantages in the defense domain. Subsequently, it elaborates on the technical implementation and superiorities of LLMs in key application scenarios, including cyber attack forensics, automated incident response, and code auditing. Finally, it examines the critical challenges encountered by LLMs in the field of cyber security and prospects future research directions. This paper profoundly reveals the core pathway and implementation mechanism underlying the paradigm shift from "passive response" to "active defense" in the cyber security domain driven by LLMs, thereby providing systematic technical references for academic research and engineering practice in this field.

Keywords

large language models (LLMs), active defense, cyber security, threat detection, intelligent security

1. Introduction

The current field of cyber defense is facing severe "asymmetry" challenges. Attackers only need to find a weak link in a system to successfully infiltrate it, while defenders have to deal with massive and ever-changing attack methods to protect the security of the entire system [1]. According to the prediction by Cybersecurity Ventures, by 2025, the global economic losses caused by cybercrime will reach 10.5 trillion US dollars, an increase of 250% compared with 3 trillion US dollars in 2015 [2]. This asymmetry is reflected in multiple aspects: attackers can make multiple attempts without being detected, while a single mistake by defenders may lead to serious consequences; attack technologies are advancing with each passing day, while traditional defense technologies lack timely and effective responses to these attacks. Traditional cyber security defense systems are mainly based on technologies such as rule matching, feature recognition, and signature detection, which are often inadequate in the face of unknown threats and zero-day attacks. Through capabilities such as

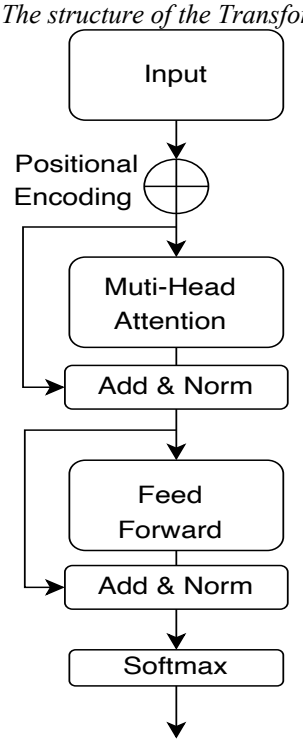
deep semantic understanding, long-range dependency modeling, and automated reasoning, LLMs provide fundamental solutions to core challenges such as unknown threat detection, intelligent analysis of massive data, and automated response decision-making. Their emergence also offers new possibilities for breaking through technical bottlenecks in more cyber defense fields, such as cyber attack forensics, automated incident response, and code auditing.

This paper aims to systematically review the technical pathways and application paradigms of large language models in the construction of active cyber security defense systems, and in-depth analyze the core mechanisms of their transformation from “passive response” to “active defense”. By sorting out the basic principles of LLMs and their unique advantages in defense scenarios, this paper focuses on exploring their technical implementation and performance in key applications such as cyber attack forensics, automated incident response, and code auditing, and objectively analyzes the current core challenges. This paper is expected to provide a technical reference for researchers and engineering practitioners in the field of cyber security, promote the standardized application and ecological construction of LLMs in active defense systems, and facilitate the leap of intelligent security technologies from theoretical exploration to practical deployment.

2. Basic Principles of Large Language Models

The Transformer architecture serves as the technical cornerstone of large language models. Its core innovation lies in being fully based on the attention mechanism, abandoning the traditional recurrent neural network (RNN) and convolutional neural network (CNN) structures [3]. This architecture mainly consists of two components: the encoder and the decoder, as shown in Figure 1. The encoder is responsible for converting the input sequence into a continuous representation, while the decoder generates the target sequence based on the output of the encoder.

Figure 1: The structure of the Transformer[3]



The core of the Transformer is the multi-head self-attention mechanism, which enables the model to simultaneously attend to all other positions in the sequence when processing each position. This allows the model to capture long-range dependencies, which is crucial for understanding complex cyber security event sequences. Positional encoding is another key component of the Transformer, which provides the model with the order information of elements in the sequence.

Large language models adopt a two-stage learning paradigm of “pre-training and fine-tuning”, which has been widely applied in models such as BERT [4]. In the pre-training phase, the model performs unsupervised learning on large-scale general corpora to acquire general language representations and knowledge. The goal of this phase is to enable the model to master basic language rules, common sense knowledge, and reasoning capabilities. In the pre-training phase, the model performs unsupervised learning on large-scale general corpora to acquire general language representations and knowledge. In the fine-tuning phase, the pre-trained model undergoes supervised fine adjustment for specific downstream tasks.

3. Applications of LLMs in the Field of Cyber Defense

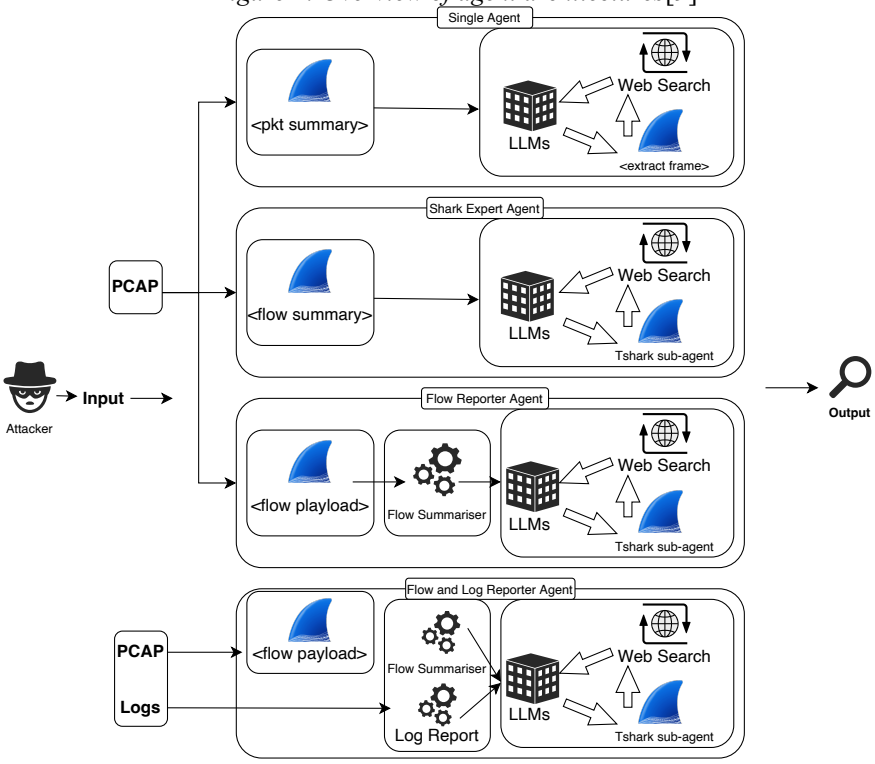
3.1 Cyber Attack Forensics

In the field of cyber attack forensics, the manual-led investigation model has long faced bottlenecks such as low efficiency, high error rate, and insufficient capability in processing large-scale traffic. With the breakthrough of LLM agent technology in the automation of complex tasks, a forensics framework integrating LLM reasoning capabilities and multi-agent collaboration has become a direction of technological innovation.

In recent years, a large number of researchers have begun to explore the use of LLMs for cyber attack forensics. Li et al. proposed an LLM-aided static analysis method, which significantly improved the accuracy and efficiency of vulnerability detection in cyber attack forensics by combining traditional static analysis tools with the reasoning capabilities of large language models [5]. Meanwhile, Liu et al. proposed the AgentFuzz framework, specifically designed to detect taint-style vulnerabilities in LLM agents, fully demonstrating the important value of LLM agents in cyber attack forensics [6].

The most representative one is CyberSleuth proposed by Fumero et al., which innovatively integrates a MemGPT-style memory management mechanism with a multi-agent collaboration architecture [7, 8], and realizes automated forensic analysis of web attack incidents through independent analysis of post-attack evidence [9]. The primary breakthrough of CyberSleuth lies in its MemGPT-style hierarchical memory management mechanism. Aiming at the problems of limited LLM context window and easy loss of focus in large-scale evidence analysis, this mechanism constructs a collaborative system of short-term memory, long-term memory, and dynamic working context. On this basis, CyberSleuth systematically designed and tested four differentiated agent architectures to verify the impact of collaboration modes on forensic performance, as shown in Figure 2.

Figure 2: Overview of agent architectures[9]



A single agent directly analyzes raw data packets. Although this design is simple, it is prone to losing analysis focus due to context window saturation when processing large-scale traffic, resulting in low traceability accuracy; the Tshark expert agent adopts a nested multi-agent structure, where the main agent coordinates specialized Tshark sub-agents to perform traffic analysis. However, problems such as ambiguous instructions and coordination failures exist between the main and sub-agents, affecting the forensic effect; the traffic report agent adopts a simple sequential arrangement and has been verified as the optimal architecture. In the preprocessing phase, sub-agents independently scan traffic, reconstruct TCP/UDP connection payloads, and apply a square root token allocation strategy to balance context consumption. In the reasoning phase, the main agent focuses on performing CVE verification and attack success assessment based on preprocessing reports, and cross-references external vulnerability databases through web search tools; although the log-enhanced agent adds a log analysis sub-agent, redundant data instead interferes with the identification of key evidence and fails to improve performance. In addition, CyberSleuth fully demonstrates its technical effectiveness through a triple verification system of "benchmark dataset testing + LLM backend comparison + human blind evaluation."

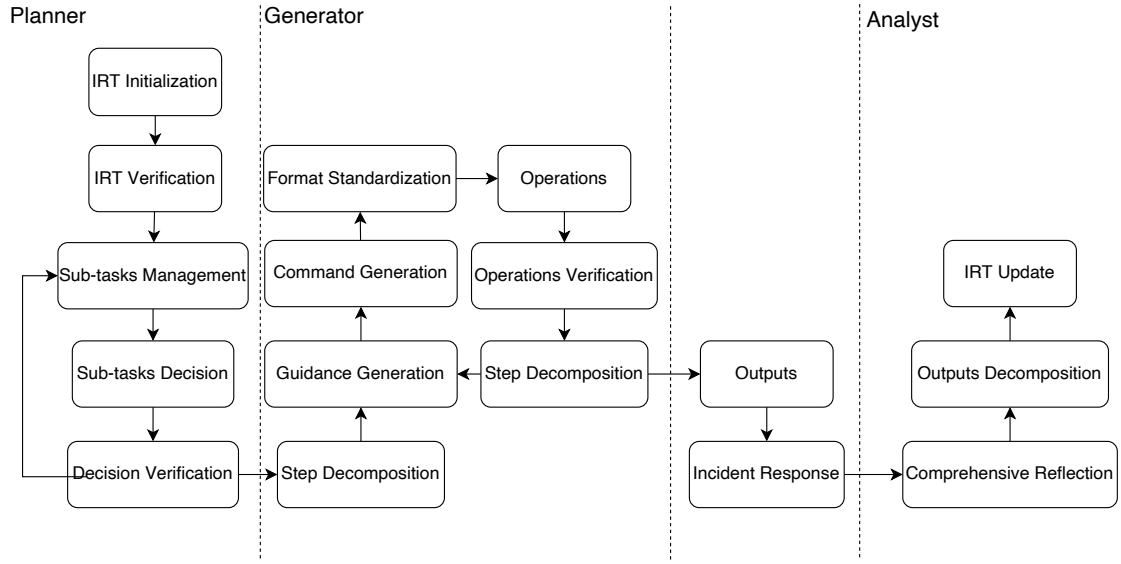
The above three methods represent different technical pathways of LLMs in the field of forensics: the method proposed by Li et al. belongs to the tool-enhanced type, which empowers traditional static analysis through LLMs and is suitable for developing security shift-left scenarios [5]; AgentFuzz proposed by Liu et al. belongs to the agent security type, whose core value lies in ensuring the security of LLM agents themselves. Although it is not a direct forensics tool, it provides a foundation for constructing trusted forensics agents [6]; CyberSleuth proposed by Fumero et al., is a system reconstruction type, which realizes end-to-end automated forensics through multi-agent collaboration, has the best verified effect in real environments, and represents the development direction of the next-generation intelligent forensics systems [9].

3.2 Automated Incident Response

In the field of cyber security, the importance of automated incident response has become increasingly prominent. LLMs are widely applied in the incident response process, covering key stages such as automated threat detection, response strategy formulation, and attack traceability and recovery. The development of these technologies provides the feasibility for addressing increasingly complex cyber attacks and significantly improves the speed and accuracy of emergency response.

In the field of automated incident response, Li et al. provided important dataset support. This dataset is specifically targeted at attention-based LLM vulnerability localization methods, laying the foundation for the reproducibility and standardized evaluation of research in this field [10]. Lin et al. systematically evaluated the capability boundaries of LLMs in real response scenarios by constructing IRBench, the first professional benchmark test for incident response [11]. Experiments show that when directly applying current top-tier models such as GPT-4, Claude-3.5, and LLaMA3-70B, the success rate of complete tasks is low. The main reasons for failure include incorrect strategy formulation, inaccurate command generation, and omission of key information. To break through these limitations, the paper innovatively proposes a four-session collaboration architecture, as shown in Figure 3.

Figure 3: Workflow of IRCOPILOT[11]



The Planner decomposes complex tasks into manageable subtasks by maintaining an incident response tree and dynamically tracks their "to-do/completed" status to address the issue of context loss; the Generator focuses on converting strategies into standardized executable instructions and strictly marks command boundaries using the "$" symbol to avoid format ambiguity; the Reflector identifies hallucinations through four-source cross-validation and triggers a rollback mechanism; the Analyst employs Tree-of-Thought technology to perform weighted analysis of multi-source attack traces. This division of responsibilities enables IRCOPILOT-GPT-4O to achieve a subtask completion rate of 136% that of the original model, while IRCOPILOT-LLaMA3-70B even achieves a 150% performance improvement, reducing the failure rate from 28.3% to 5.4%. In addition, in 5 real attack cases, IRCOPILOT can accurately extract attacker IPs, Webshell connection parameters, and malicious process trees, with the average response cost controlled in the range of 0.25 to 0.90 US dollars.

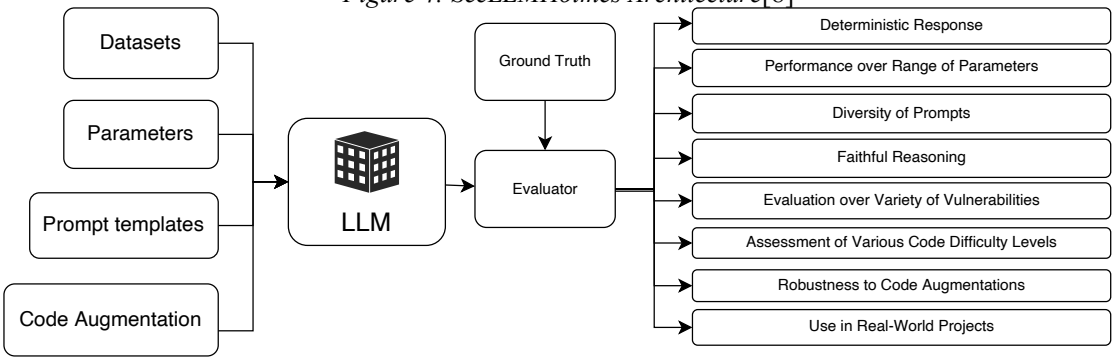
Li et al.'s dataset addresses the pain point of the lack of standardized evaluation in domain research, providing a research foundation for subsequent methods [10]; although native LLMs have strong versatility, they expose the limitation of rigid application in professional scenarios, with failures mainly concentrated in three dimensions: strategy, command, and information; IRCOPILOT achieves a qualitative leap through professional division of responsibilities and a reflective verification mechanism. Its key insight is that complex tasks should not rely on a single model to complete end-to-end, but should improve reliability through multi-expert collaboration and quality loop control [11].

3.3 Code Auditing

Against the backdrop of the continuous escalation of software supply chain security threats, code auditing, as the core defense line for ensuring the intrinsic security of software, is facing the dual challenges of large-scale analysis and precise identification. With their capabilities in deep code understanding and cross-context semantic reasoning, large language models are reconstructing the technical paradigm of traditional static analysis, demonstrating intergenerational advantages in scenarios such as vulnerability detection, automated repair, and smart contract auditing.

Ullah et al. constructed a dedicated vulnerability analysis framework based on GPT-3.5 fine-tuning, as shown in Figure 4. SecLLMHolmes is a fully automated framework for systematically evaluating the vulnerability detection capabilities of LLMs [8]. Its architecture consists of three layers: the input layer generates diverse test inputs through 228 code scenarios and 17 prompt templates; the processing layer integrates 8 mainstream LLMs and fixes temperature=0.0 to ensure determinism; the evaluation layer automatically extracts binary answers and 100-word reasoning summaries, and compares them with manually annotated ground truth through three metrics: Rouge score, cosine similarity, and GPT-4 alignment verification, quantifying performance from eight dimensions: determinism, parameter sensitivity, prompt robustness, reasoning faithfulness, vulnerability type coverage, code difficulty generalization, resistance to enhancement disturbances, and applicability in real scenarios. To break through the limitations of traditional tools in vulnerability correlation tracking, this framework innovatively introduces a code semantic flow graph and a cross-function data flow tracking mechanism, which can accurately depict code dependency relationships and data propagation paths related to vulnerabilities. Evaluation results on the SARD standard vulnerability dataset show that its vulnerability detection F1 score reaches as high as 92.65%, significantly outperforming classic deep learning vulnerability detection baseline models such as SyseVR and VulDeBERT, and filling the gap of traditional detection schemes in identifying complex logical vulnerabilities.

Figure 4: SecLLMHolmes Architecture[8]



Mhatre et al. took three mainstream LLMs, namely ChatGPT-4, Claude 3, and LLaMA 4, as core tools, constructed a multi-type vulnerability auditing benchmark dataset covering C++ and Python, designed a multi-stage context-aware prompt protocol, and adopted a hierarchical system to evaluate auditing effects from three dimensions: detection accuracy, reasoning depth, and repair quality [12]. Relevant data show that all models perform excellently in basic code auditing, achieving a 100% complete detection rate for 7 types of common vulnerabilities among C++ beginners, and can accurately provide repair solutions that comply with modern specifications. However, their performance differentiates significantly in complex scenarios. ChatGPT-4 and Claude 3 can not only identify buffer overflows for C++ format string vulnerabilities but also correlate shell command injection risks, achieving a complete detection rate of over 80% for the OpenSSL EVP_PKEY_assign null pointer vulnerability. In contrast, LLaMA 4 misses file descriptor issues when auditing environment variable manipulation vulnerabilities, fails to distinguish the semantic difference between “tuple single key/multi-key” for Pandas MultiIndex grouping vulnerabilities, and its complete detection rate is less than 50%. In addition, only ChatGPT-4 can fully detect and repair Python’s pathlib. Path type error vulnerabilities. Moreover, the average compliance rate of repair suggestions of all models in basic vulnerability auditing is 92%, which is significantly higher than 68% for complex production-level vulnerabilities, verifying the value and current limitations of LLMs as the first-round review tool for code auditing. Sun et al.’s research on integrating large language models for code vulnerability detection also experimentally proved that ensemble learning methods can significantly improve the performance of LLMs in vulnerability detection, especially performing excellently in handling class imbalance and multi-class classification tasks [13].

The above three methods represent different technical routes for LLM-based code auditing: SecLLMHolmes takes the path of dedicated model optimization, which is suitable for scenarios requiring extremely high accuracy [14]; the research by Mhatre et al. belongs to the benchmark evaluation type, whose value lies in clarifying the capability boundaries of LLMs and providing a selection basis for practical

deployment [12]; Sun's ensemble learning explores a performance-enhanced path, offering ideas for addressing the problem of unbalanced vulnerability distribution in the real world [13].

4. Challenges and Prospects

4.1 Key Challenges

The implementation of large language models in active defense systems faces three major structural contradictions: First, the dilemma of cost-precision trade-off. Although the commercial GPT-5 achieves a 90% service recognition rate in CyberSleuth, the cost per analysis is as high as 8.63 US dollars, restricting large-scale deployment [5]; although open-source models reduce the cost to 0.29-0.54 times, their CVE recognition rate is generally lower than 70%, forming an economy-effectiveness paradox. Second, the bottleneck of cognitive reliability. Empirical evidence from IRCOPILOT shows that the failure rate of native LLMs in 130 subtasks reaches 28.3%, and the hallucination problem leads to a strategy formulation error rate of over 40%. Even with the introduction of a four-source cross-validation mechanism, context collapse still causes 15%-20% of key forensic evidence to be lost in traffic analysis, highlighting the problem of semantic drift [6]. Third, adversarial vulnerability and security alignment risks, coupled with the lack of differential privacy guarantees for the training and use of sensitive security data. Adversarial machine learning attacks may cause the model to make wrong predictions by adding carefully designed perturbations to the model input, which poses a serious threat to the application of LLMs in the field of cyber security [15].

4.2 Future Research Directions

It is urgent to construct a vertical domain adaptive architecture, achieve F1 > 92% on the SARD dataset through parameter-efficient fine-tuning such as LoRA, and simultaneously explore model quantization and distillation technologies to solve the cost dilemma [14]. Efforts should be made to deepen the multi-agent collaboration mechanism, optimize the CyberSleuth-style hierarchical memory management, introduce dynamic role assignment and symbolic reasoning engines, and enhance the causal traceability capability of complex attack chains [9]. Ultimately, it is necessary to establish an interpretable attack-defense evaluation framework, formulate a unified security capability benchmark test protocol, integrate adversarial training and formal verification, promote the formation of a regulatory ecosystem covering data privacy, model robustness, and emergency response standards, and realize the paradigm shift from black-box detection to trusted defense.

5. Conclusions

The great application value and potential of LLMs in the field of cyber defense are driving a fundamental transformation of the cyber security paradigm from "passive response" to "active defense". Through capabilities in deep semantic understanding and automated reasoning, LLMs have not only broken through the technical bottlenecks of traditional rule-based detection but also achieved intelligent upgrading in core scenarios such as attack forensics, incident response, and code auditing. However, the current application of LLMs still faces multiple challenges such as cost-precision trade-off, cognitive reliability bottlenecks, and adversarial vulnerability. The complexity of these challenges goes far beyond the technical level, involving systematic issues such as data privacy, model security, and ethical governance. Therefore, while promoting the application of these new technologies, a cautious and rational attitude must be maintained. It is urgent to establish a full-process regulatory mechanism covering data annotation, model evaluation, and emergency response, and formulate unified security capability benchmark test standards. Future research should focus on directions such as parameter-efficient fine-tuning, multi-agent collaboration optimization, and interpretability enhancement, promoting the formation of a new "human-machine collaboration" defense ecosystem. Only by strengthening in-depth interdisciplinary and cross-domain cooperation and building an industry-university-research-application collaborative innovation system can people ensure the trusted development and sustainable application of LLMs in the field of cyberspace security, and ultimately realize the practical deployment and large-scale implementation of intelligent defense technologies.

References

[1] Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z. and Zhang, Y. A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. High-Confidence Computing. 2024, 4(2), p. 100211. https://doi.org/10.1016/j.hcc.2024.100211.

[2] Morgan, S. Cybercrime To Cost The World $10.5 Trillion Annually By 2025. Available from: https://cybersecurityventures.com/cyberwarfare-report-intrusion (accessed 8 January 2026).

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. Attention is all you need. In Advances in neural information processing systems, New York, 2017; pp. 6000-6010.

[4] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), Minneapolis, Minnesota, USA, 2019; pp. 4171-4186.

[5] Li, Z., Dutta, S. and Naik, M. IRIS: LLM-assisted static analysis for detecting security vulnerabilities. arXiv preprint arXiv:2405.17238. 2024. https://doi.org/10.48550/arXiv.2405.17238.

[6] Liu, F., Zhang, Y., Luo, J., Dai, J., Chen, T., Yuan, L., Yu, Z., Shi, Y., Li, K. and Zhou, C. Make agent defeat agent: Automatic detection of {Taint-Style} vulnerabilities in {LLM-based} agents. In 34th USENIX Security Symposium (USENIX Security 25), Seattle, Washington, USA, 2025; pp. 3767-3786.

[7] Packer, C., Fang, V., Patil, S. G., Lin, K., Wooders, S. and Gonzalez, J. E. MemGPT: Towards LLMs as Operating Systems. arXiv preprint arXiv:2310.08560. 2023. https://doi.org/10.48550/arXiv.2310.08560.

[8] Tran, K.-T., Dao, D., Nguyen, M.-D., Pham, Q.-V., O'Sullivan, B. and Nguyen, H. D. Multi-agent collaboration mechanisms: A survey of llms. arXiv preprint arXiv:2501.06322. 2025. https://doi.org/10.48550/arXiv.2501.06322.

[9] Fumero, S., Huang, K., Boffa, M., Giordano, D., Mellia, M., Houidi, Z. B. and Rossi, D. CyberSleuth: Autonomous Blue-Team LLM Agent for Web Attack Forensics. arXiv preprint arXiv:2508.20643. 2025. https://doi.org/10.48550/arXiv.2508.20643.

[10] Li, Y., Li, X., Wu, H., Zhang, Y., Cheng, X., Zhong, S. and Xu, F. Attention is all you need for llm-based code vulnerability localization. IACAPAP ArXiv (Online). 2024. https://doi.org/10.48550/arxiv.2410.15288.

[11] Lin, X., Zhang, J., Deng, G., Liu, T., Liu, X., Yang, C., Zhang, T., Guo, Q. and Chen, R. IRCopilot: Automated Incident Response with Large Language Models. arXiv preprint arXiv:2505.20945. 2025. https://doi.org/10.48550/arXiv.2505.20945.

[12] Mhatre, A., Nader, N., Diehl, P. and Gupta, D. Llm-guard: Large language model-based detection and repair of bugs and security vulnerabilities in c++ and python. arXiv preprint arXiv:2508.16419. 2025. https://doi.org/10.48550/arXiv.2508.16419.

[13] Sun, Z., Li, J., Wan, Y., Li, C., Zhang, H., Li, G., Liu, H., Lyu, C. and Hu, S. Ensembling Large Language Models for Code Vulnerability Detection: An Empirical Evaluation. arXiv preprint arXiv:2509.12629. 2025. https://doi.org/10.48550/arXiv.2509.12629.

[14] Ullah, S., Han, M., Pujar, S., Pearce, H., Coskun, A. and Stringhini, G. Can large language models identify and reason about security vulnerabilities? not yet. arXiv preprint arXiv:2312.12575. 2023.

[15] Biggio, B. and Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 2018; pp. 2154-2156.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.