

Review Article

The Security Duality of Large Language Model Agents: A Systematic Review of Self-Security Protection and Cybersecurity Empowerment

Dasen Fu^{1,*} , Dalin Xiang² 

¹Department of Electrical Engineering and Automation, Hainan International Commercial Aerospace Launch Co., Ltd, Wenchang, China

²Department of General Mechanics and Foundations of Mechanics, Hainan International Commercial Aerospace Launch Co., Ltd, Wenchang, China

Abstract

Large Language Model (LLM) agents integrate memory, tool invocation, environment interaction, and multi-agent collaboration, evolving from passive text-generation tools into autonomous systems that execute complex tasks. This capability leap creates a pronounced security duality. On the one hand, LLM agents face emerging threats such as jailbreak attacks, prompt injection, memory poisoning, and multi-agent collusion, requiring systematic self-security protection. On the other hand, they empower cybersecurity in vulnerability detection, penetration testing, threat intelligence, and malicious code analysis, while their autonomous attack capabilities, e.g., zero-day exploitation, raise concerns about malicious abuse. Existing reviews often treat these two lines separately. This review systematically examines key studies from 2023 to 2026, covering attacks on agents and agent-enabled cyber operations. We propose a unified analytical framework with two dimensions: Dimension 1, self-security protection, and Dimension 2, cybersecurity empowerment. We organize self-security threats into model-layer safety alignment failure and fine-tuning-induced safety weakening; prompt-layer jailbreak and indirect prompt injection; memory/knowledge-layer retrieval-augmented generation (RAG) poisoning and long-term memory contamination; and multi-agent-layer prompt infection and secret collusion. For cybersecurity empowerment, we review vulnerability detection and remediation, automated penetration testing, cyber threat detection, autonomous vulnerability exploitation, and information ecosystem security. We further identify four coupling mechanisms: security weaknesses as attack entry points, tool integration expanding the attack surface, bidirectional capability enhancement, and double-edged multi-agent architectures. Finally, we distill five challenges—security-capability parity, native security design, multi-agent governance, ecosystem-oriented evaluation, and responsible capability release—and outline future directions for responsible development and deployment of LLM agents.

Keywords

Large Language Model Agents, Security Protection, Cybersecurity, Jailbreak Attacks, Prompt Injection, Vulnerability Detection

*Correspondence: Dasen Fu (15091994360@163.com), Dasen Fu (709366545@qq.com)

Received: 11 September 2026; **Accepted:** 20 September 2026; **Published:** 30 September 2026



Copyright: © The Author(s), 2026. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

1. Introduction

With the rapid iteration of large language models such as GPT-4, Claude, and Llama, LLM agents are emerging as a new-generation paradigm of AI systems. Unlike traditional LLMs that only handle text input and output, agents acquire the ability to plan, reason, and act autonomously by introducing mechanisms such as memory modules (long-term memory, retrieval-augmented generation), tool invocation (APIs, code execution), environment perception, and multi-agent collaboration [1]. However, this architectural expansion introduces security risks far beyond those of traditional LLMs. In their KDD 2025 survey, Yu et al. proposed the TrustAgent framework, dividing the trust dimensions of agents into intrinsic aspects (brain, memory, tools) and extrinsic aspects (user, agent, environment) [1]. He et al. further analyzed the full spectrum of threats facing LLM agents from the dual perspectives of security and privacy [2]. Meanwhile, Das et al. conducted a systematic survey of security and privacy challenges for LLMs as a whole, pointing out that jailbreak attacks, data poisoning, and personal information leakage are the three core risks [3].

At the same time, the application of LLM agents in the cybersecurity domain is experiencing explosive growth. Zhang et al. systematically analyzed more than 300 cross-

disciplinary studies on LLMs and cybersecurity, covering 25 LLMs and more than 10 downstream scenarios [4]. Xu et al., from the perspective of software security, analyzed 185 papers from top security conferences and found that LLM agents are shifting from single-task execution to orchestrating complex multi-step security workflows [5].

The core argument of this paper is that the security duality of LLM agents—namely, the dual roles of "victim under attack" and "enabler/attacker in cybersecurity"—constitutes a tightly coupled unity of opposites. Weak links in security protection may be precisely what malicious actors exploit, while advances in security empowerment in turn deepen their own attack surface. Understanding this duality is crucial for the responsible development and deployment of LLM agents.

2. Analytical Framework for the Security Duality of LLM Agents

Based on existing literature, we propose a unified "security duality" analytical framework that categorizes the security issues of LLM agents into two major dimensions. See Table 1.

Table 1. categorizes the security issues of LLM agents into two major dimensions.

Dimension	Core Question	Typical Threats/Applications	Representative Works
Dimension 1: Self-Security Protection	How can agents defend against external attacks?	Jailbreak attacks, prompt injection, memory poisoning, multi-agent collusion	[6-9]
Dimension 2: Cybersecurity Empowerment	How can agents be applied to the security domain?	Vulnerability detection, penetration testing, threat detection, automated remediation, (malicious) vulnerability exploitation	[4, 5, 10-12]

These two dimensions are not independent of each other. Vulnerabilities in security protection may be exploited by attackers to carry out cyberattacks (e.g., jailbreaking an agent to generate malicious code), while advances in security empowerment, in turn, expand the agent's own attack surface (e.g., more tool-invocation interfaces mean more entry points for prompt injection).

3. Dimension 1: Self-Security Protection of LLM Agents

3.1. Failure of Safety Alignment: Jailbreak Attacks

LLMs are typically trained with safety alignment (e.g.,

RLHF) before deployment, but Wei et al., in their pioneering work at NeurIPS 2023, revealed two failure modes of safety training: Competing Objectives and Mismatched Generalization [6]. The former refers to conflicts between model capabilities and safety objectives, while the latter refers to safety training failing to cover certain domains in which the model already possesses capabilities. The study demonstrated that even GPT-4 and Claude, after extensive red-teaming and safety training, can still be successfully jailbroken. Qi et al., in their ICLR 2024 study, revealed an even more concerning phenomenon: fine-tuning GPT-3.5 Turbo with only 10 adversarial samples (at a cost of less than \$0.20) can completely bypass its safety protections [13]. More seriously, even fine-tuning with benign datasets can unintentionally weaken safety alignment. This finding indicates that current safety infrastructure has fundamental flaws in coping with custom fine-tuning scenarios.

3.2. Prompt Injection Attacks: The "Code Execution" of the Agent Era

When LLMs expand from closed dialogue systems into agents integrated with external data sources, indirect prompt injection becomes the most representative new type of attack. Abdelnabi et al. were the first to systematically reveal this attack vector: by embedding malicious prompts in external data sources (such as web pages and documents), attackers can remotely hijack the functionality of LLM-integrated applications without the user's knowledge [7]. The study verified the feasibility of the attack in real systems such as Bing Chat and code completion engines, demonstrating severe consequences including data theft, worm propagation, and contamination of the information ecosystem. The AgentDojo evaluation framework proposed by DeBenedetti et al. further quantified agents' vulnerability to prompt injection [14]. The framework contains 97 real tasks (such as email management, online banking operations, and travel booking) and 629 security test cases, and found that even state-of-the-art LLMs fail on many tasks even under no-attack conditions, while existing prompt injection attacks, although capable of breaching some security properties, cannot yet achieve comprehensive compromise.

3.3. Memory and Knowledge Base Poisoning

LLM agents typically rely on memory modules or retrieval-augmented generation (RAG) mechanisms to leverage historical experience and external knowledge. AgentPoison, proposed by Chen et al., is the first backdoor attack method against RAG agents [8]. By injecting optimized backdoor triggers into the knowledge base or long-term memory, this method causes user instructions containing the trigger to retrieve malicious

demonstrations with high probability, while benign instructions remain unaffected. On three types of real agents—autonomous driving, knowledge question answering, and medical electronic health records—AgentPoison achieved an average attack success rate exceeding 80%, with an impact on benign performance below 1% and a poisoning rate below 0.1%.

3.4. Security Threats in Multi-Agent Systems

Multi-agent systems (MAS) introduce more complex security dimensions. The Prompt Infection attack proposed by Lee et al. reveals the self-propagating nature of prompt injection in multi-agent environments [15]. In this attack, malicious prompts can self-replicate among interconnected agents, behaving like a computer virus, leading to data theft, fraud, and system-level paralysis. Experiments show that multi-agent systems are highly susceptible to such attacks, and even if agents do not openly share all communication content, they are not immune. More covertly, Motwani et al., at NeurIPS 2024, first formalized the problem of Secret Collusion among AI agents [9]. Two or more agents can use steganography to hide the true nature of their interactions and evade oversight. The study provides rigorous theoretical analysis and empirically demonstrates the growing steganographic capabilities of frontier models, revealing the limitations of countermeasures such as monitoring, paraphrasing, and parameter optimization.

3.5. Summary: Hierarchy of Challenges in Self-Security Protection

Based on the above studies, the challenges in the self-security protection of LLM agents can be summarized into the following hierarchy, See Table 2.

Table 2. The challenges in the self-security protection of LLM agents can be summarized into the following hierarchy.

Attack Layer	Core Threat	Representative Works
Model layer	Safety alignment failure, fine-tuning weakening safety	[6, 13]
Prompt layer	Jailbreak attacks, indirect prompt injection	[6, 7]
Memory/Knowledge layer	RAG poisoning, long-term memory contamination	[8]
Multi-agent layer	Prompt infection propagation, secret collusion	[9, 15]

4. Dimension 2: LLM Agents Empowering Cybersecurity

4.1. Vulnerability Detection and Remediation

The application of LLMs in the field of software

vulnerability detection and remediation is the most mature direction of cybersecurity empowerment. Pearce et al., at IEEE S&P 2023, were the first to study the zero-shot vulnerability remediation capability of LLMs [12], verifying that models such as Codex and Jurassic can remediate 100% of vulnerabilities in synthetic and handcrafted scenarios, but functional correctness on real historical vulnerabilities remains challenging. In a recent survey in ACM Computing Surveys, Sheng et

al. systematically analyzed problem modeling, model selection, application methods, and evaluation metrics of LLMs in vulnerability detection [16]. The study pointed out that cross-language detection, multimodal integration, and repository-level analysis are the main current challenges, and proposed solutions for dataset scalability, model interpretability, and low-resource scenarios. SEC-bench proposed by Lee et al. is the first automated evaluation framework for real-world software security tasks [17]. The framework adopts a multi-agent architecture to automatically build code repositories, reproduce vulnerabilities, and generate patches. Evaluation shows that current state-of-the-art LLM code agents achieve only an 18.0% success rate in PoC generation and 34.0% in vulnerability remediation, indicating that LLM agents still have substantial room for improvement in real-world security engineering.

4.2. Automated Penetration Testing

PentestAgent proposed by Shen et al. introduces LLM agents into automated penetration testing [18]. The framework uses multi-agent collaboration to automate intelligence gathering, vulnerability analysis, and exploitation phases, enhancing penetration testing knowledge through RAG and reducing manual intervention. Comprehensive benchmarks show that it outperforms existing methods in task completion and efficiency.

4.3. Cyber Threat Detection

SecurityBERT proposed by Ferrag et al. demonstrates the potential of LLMs in IoT network threat detection [19]. By combining privacy-preserving encoding and a Byte-level BPE tokenizer, SecurityBERT achieved 98.2% accuracy on the Edge-IIoTset dataset, covering 14 attack types, with inference time below 0.15 seconds, making it suitable for deployment on resource-constrained IoT devices. The systematic review by Xu et al. further confirmed the broad application of LLMs in fields such as vulnerability detection, malware analysis, and network intrusion detection [5], and observed that LLM-based autonomous agents represent a paradigm shift from single-task execution to orchestrating complex multi-step security

workflows.

4.4. Autonomous Attack Capabilities of LLM Agents: The Other Side of the Double-Edged Sword

Beyond the positive applications of empowering cybersecurity, the autonomous attack capabilities demonstrated by LLM agents have raised serious concerns. Fang et al. demonstrated that a GPT-4 agent can autonomously exploit one-day vulnerabilities given a CVE description, with a success rate significantly better than GPT-3.5 and traditional vulnerability scanners [10]. Among 15 real vulnerabilities (including multiple severity levels), GPT-4 successfully exploited 87% of the test cases. Zhu et al. further pushed this capability into the domain of zero-day vulnerabilities [11]. Their proposed HPTSA system uses a planning agent to dispatch sub-agents, addressing the shortcomings of a single agent in exploring multiple vulnerabilities and long-horizon planning, and improved performance by 4.3 times on a benchmark of 14 real zero-day vulnerabilities. These studies directly highlight the security risks of widely deploying highly capable LLM agents.

4.5. Information Ecosystem Security

Pan et al. revealed another threat of LLMs from the perspective of information pollution [20]. The study showed that LLMs can serve as efficient misinformation generators, causing performance degradation of open-domain question-answering systems by up to 87%. More alarmingly, the attributes that influence human judgment differ from those that influence machine judgment, which places human-centered anti-misinformation strategies in a fundamental dilemma when facing LLM-generated misinformation.

4.6. Summary: The Bidirectionality of Security Empowerment

Research on cybersecurity empowerment reveals the bidirectional instrumentality of LLM agents. See Table 3.

Table 3. Research on cybersecurity empowerment reveals the bidirectional instrumentality of LLM agents.

Direction	Application Scenario	Risk/Benefit
Defensive Empowerment	Vulnerability detection, automated remediation, threat detection, penetration testing	Improves security operations efficiency and reduces labor costs
Offensive Abuse	Autonomous vulnerability exploitation, misinformation generation, social engineering	Lowers the barrier to attack and expands the scale of attacks

5. Coupling Relationships of Security Duality

The above two dimensions are not simply parallel but deeply coupled. We identify the following coupling mechanisms:

(1) Security weaknesses as attack entry points. Jailbreak attacks and prompt injection not only endanger the agent's own security but can also be exploited to drive agents to perform malicious cyber operations. For example, attackers can jailbreak an agent to generate exploit code or execute unauthorized system operations.

(2) Tool integration expands the attack surface. The tools that agents integrate to empower cybersecurity (code execution, file access, network requests) precisely provide more entry points for prompt injection. AgentDojo's evaluation shows that the more tool invocations, the larger the attack surface [14].

(3) Bidirectional effects of capability enhancement. Improvements in LLM agents' reasoning and planning capabilities enhance both their security analysis capabilities (e.g., more accurate vulnerability detection) and their autonomous attack capabilities (e.g., zero-day exploitation [11]). The "arms race" between security capabilities and attack capabilities essentially shares the same technological foundation.

(4) Double-edged effects of multi-agent architectures. Multi-agent collaboration improves the level of automation in security operations (e.g., PentestAgent [18]), but also introduces entirely new threats such as prompt infection propagation [15] and secret collusion [9].

6. Current Challenges and Future Directions

Challenge 1: Security-capability parity. Wei et al. emphasized that safety mechanisms should be as complex as the underlying models [6]. Current safety alignment lags far behind the growth of model capabilities, and safety protections can be easily weakened, especially in custom fine-tuning scenarios [13]. In the future, it is necessary to establish a security protection system covering the entire lifecycle of training, fine-tuning, and deployment.

Challenge 2: Native security design for agent architectures. Existing defenses are mostly post hoc patches and lack methods for embedding security into agent design at the architectural level. The TrustAgent framework [1] provides a modular taxonomy, but how to translate security constraints into executable architectural components remains to be explored.

Challenge 3: Multi-agent security governance. There are no mature solutions for threats such as prompt infection [15] and secret collusion [9] in multi-agent systems. Preliminary defense mechanisms such as LLM Tagging need to be combined with existing security measures, but still face challenges in scalability and robustness.

Challenge 4: Ecosystem-oriented security evaluation. AgentDojo [14] and SEC-bench [17] represent important advances in evaluation frameworks, but existing evaluations still struggle to cover the complexity and ambiguity of the real world. It is necessary to build a dynamic evaluation ecosystem covering the full spectrum of attack-defense.

Challenge 5: Responsible capability release. Research on autonomous vulnerability exploitation [10, 11] shows that the attack capabilities of LLM agents are rapidly approaching the level of human security experts. How to promote defensive applications while managing the risk of offensive abuse requires collaborative governance by the technical community, policymakers, and industry.

7. Conclusion

This paper systematically reviews the research landscape of the security duality of large language model agents. In the dimension of self-security protection, jailbreak attacks, prompt injection, memory poisoning, and multi-agent collusion constitute a multi-level threat chain from the model layer to the system layer, and current safety alignment mechanisms are increasingly fragile in the face of capability growth. In the dimension of cybersecurity empowerment, LLM agents demonstrate significant value in scenarios such as vulnerability detection, penetration testing, and threat intelligence, but their autonomous attack capabilities also sound a security alarm. The deep coupling of the two dimensions—security weaknesses as attack entry points, tool integration expanding the attack surface, and the bidirectional effects of capability enhancement—requires us to adopt a holistic perspective in examining the security governance of LLM agents. Future research should focus on four major directions: security-capability parity, native security at the architectural level, multi-agent governance, and dynamic evaluation ecosystems, so as to achieve the responsible development of LLM agents.

Abbreviations

LLM	Large Language Model
AI	Artificial Intelligence
GPT	Generative Pre-trained Transformer

Author Contributions

Dasen Fu: Conceptualization, Resources, Supervision, Writing – original draft, Writing – review & editing

Dalin Xiang: Data curation, Investigation, Methodology, Validation, Visualization

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Yu M, Meng F, Zhou X, et al. A Survey on Trustworthy LLM Agents: Threats and Countermeasures [C]. KDD 2025. <https://doi.org/10.1145/3690624.3709179>
- [2] He F, Zhu T, Ye D, et al. The Emerged Security and Privacy of LLM Agent: A Survey with Case Studies [J]. ACM Computing Surveys, 2026. <https://doi.org/10.1145/3773080>
- [3] Das B C, Amini M H, Wu Y. Security and Privacy Challenges of Large Language Models: A Survey [J]. ACM Computing Surveys, 2025. <https://doi.org/10.1145/3712001>
- [4] Zhang J, Bu H, Wen H, et al. When LLMs Meet Cybersecurity: A Systematic Literature Review [J]. Cybersecurity, 2025. <https://doi.org/10.1186/s42400-025-00361-w>
- [5] Xu H, Wang S, Li N, et al. Large Language Models for Cyber Security: A Systematic Literature Review [J]. ACM Transactions on Software Engineering and Methodology, 2025. <https://doi.org/10.1145/3714464>
- [6] Wei A, Haghtalab N, Steinhardt J. Jailbroken: How Does LLM Safety Training Fail?[C]. NeurIPS 2023. <https://doi.org/10.48550/arXiv.2307.02483>
- [7] Abdelnabi S, Greshake K, Mishra S, et al. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection [C]. AISEC 2023. <https://doi.org/10.1145/3605764.3623985>
- [8] Chen Z, Xiang Z, Xiao C, et al. AgentPoison: Red-teaming LLM Agents Via Poisoning Memory or Knowledge Bases [C]. NeurIPS 2024. <https://doi.org/10.48550/arXiv.2407.12784>
- [9] Motwani S R, Baranchuk M, Strohmeier M, et al. Secret Collusion among AI Agents: Multi-Agent Deception Via Steganography [C]. NeurIPS 2024. <https://doi.org/10.48550/arXiv.2402.07510>
- [10] Fang R, Bindu R, Gupta A, et al. LLM Agents Can Autonomously Exploit One-day Vulnerabilities [J]. arXiv preprint, 2024. <https://doi.org/10.48550/arXiv.2404.08144>
- [11] Zhu Y, Kellermann A, Gupta A, et al. Teams of LLM Agents Can Exploit Zero-Day Vulnerabilities [C]. EACL 2026. <https://doi.org/10.48550/arXiv.2406.01637>
- [12] Pearce H, Tan B, Ahmad B, et al. Examining Zero-Shot Vulnerability Repair with Large Language Models [C]. IEEE S&P 2023. <https://doi.org/10.1109/SP46215.2023.10179324>
- [13] Qi X, Zeng Y, Xie T, et al. Fine-tuning Aligned Language Models Compromises Safety, Even when Users Do Not Intend To [C]. ICLR 2024. <https://doi.org/10.48550/arXiv.2310.03693>
- [14] Debenedetti E, Zhang J, Balunovic M, et al. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents [C]. NeurIPS 2024. <https://doi.org/10.48550/arXiv.2406.13352>
- [15] Lee D, Tiwari M, Miranda B. Prompt Infection: LLM-to-LLM Prompt Injection Within Multi-Agent Systems [C]. ESORICS 2025 Workshops. <https://doi.org/10.48550/arXiv.2410.07283>
- [16] Sheng Z, Chen Z, Gu S, et al. LLMs in Software Security: A Survey of Vulnerability Detection Techniques and Insights [J]. ACM Computing Surveys, 2026. <https://doi.org/10.1145/3708535>
- [17] Lee H, Zhang Z, Lu H, et al. SEC-bench: Automated Benchmarking of LLM Agents on Real-World Software Security Tasks [C]. NeurIPS 2025. <https://doi.org/10.48550/arXiv.2506.11791>
- [18] Shen X, Wang L, Li Z, et al. PentestAgent: Incorporating LLM Agents to Automated Penetration Testing [C]. ACM AsiaCCS 2025. <https://doi.org/10.1145/3708821.3733879>
- [19] Ferrag M A, Ndhlovu M, Tihanyi N, et al. Revolutionizing Cyber Threat Detection with Large Language Models [J]. arXiv preprint, 2023. <https://doi.org/10.48550/arXiv.2306.14263>
- [20] Pan Y, Pan L, Chen W, et al. On the Risk of Misinformation Pollution with Large Language Models [C]. EMNLP 2023. <https://doi.org/10.18653/v1/2023.findings-emnlp.139>