

Security of Multi-Agent LLM Pipelines: Towards an Integrated Taxonomy of Inter-Agent Attacks, Cascading Risks and Defence Mechanisms

Krish Sen

Student

See photos Map of RGPV Rajiv Gandhi Proudhyogiki Vishwavidyalaya

Abstract

The rapid emergence of agentic artificial intelligence has introduced multi-agent LLM pipelines capable of autonomous planning, communication, tool use and delegated decision-making. However, these interconnected capabilities create security risks that extend beyond individual agent vulnerabilities. This paper critically examines existing research on attacks and defence mechanisms in multi-agent LLM systems, with particular emphasis on inter-agent communication, memory, tool use, trust propagation and cascading compromise. It identifies limitations in existing security taxonomies and proposes an integrated analytical perspective linking attack surfaces, propagation mechanisms, system impacts and defence layers. This article argues for the development of architecture-aware, empirically validated security frameworks for multi-agent AI. Rather than proposing another general taxonomy of agentic AI attacks, this paper focuses on the underexamined relationship between attack initiation, inter-agent propagation, system-level impact and defence containment. It offers an initial literature-based framework for assessing the potential diffusion of local compromises through a network of interconnected agents, memory, communication channels and tool interfaces. This pipeline-oriented perspective thus complements existing taxonomies by emphasizing the dynamics in terms of mutually reinforcing mechanisms of compromise and containment.

Keywords: Agentic AI; Large Language Models; Multi-Agent Systems; Cybersecurity; Inter-Agent Communication; Prompt Injection; Attack Taxonomy; Defence Mechanisms; Cascading Attacks; AI Security.

1. Introduction

Large language models (LLMs) have advanced astonishingly quickly, taking AI from answering single questions to modern more generic systems capable of planning, both reasoning and external tool use enable LLMs to perform multi-step operations. This move paved the way toward agentic AI, where LLM-based agents can independently or with sparse autonomy interpret goals, make decisions and interact with the external environment. Multi-agent LLM systems build on this by splitting tasks across multiple specialized agents that coordinate over a predefined workflow in order to communicate, allocate actions and exchange knowledge. While such architectures can improve scalability, specialisation and task performance, they also introduce security risks that are not adequately represented by security models designed for conventional LLM applications or isolated agents (Gan et al., 2026; Tang et al., 2026).

The security challenge arises partly from the interconnected nature of multi-agent pipelines. An individual agent may interact with users, retrieval systems, memory stores, external tools and other agents, creating multiple points at which malicious instructions or compromised information can enter and propagate through the system. Research on indirect prompt injection has demonstrated that untrusted external content can manipulate LLM-integrated applications and influence subsequent model behaviour (Abdelnabi et al., 2023; Zhan et al., 2024). In an agentic environment, however, the consequences may extend beyond the initially targeted component because an agent can pass manipulated information or instructions to another agent, invoke privileged tools or contribute to subsequent decisions. Consequently, a relatively local compromise may develop into a broader system-level failure.

Recent scholarship has begun to establish taxonomies of attacks and defence mechanisms for LLM-based agents. For example, existing surveys classify threats involving prompts, tools, memory, environmental interaction and other agent components, while emerging benchmarks provide systematic approaches for evaluating attacks and defences (Gan et al., 2026; Tang et al., 2026; Zhang et al., 2025). Nevertheless, these contributions remain fragmented when considered from the perspective of multi-agent pipelines. In particular, insufficient attention has been given to systematically connecting the **source of an attack, the targeted component, the inter-agent propagation mechanism, the resulting system impact and the corresponding defence layer**. Recent research on agent workflows

This distinction is important because security in multi-agent systems cannot be assessed exclusively by testing whether individual agents resist known attacks. Agents operate within dynamic networks of trust, delegation and information exchange. A compromised agent may therefore become an attack vector against otherwise secure agents, particularly when information is implicitly trusted or when agents possess different levels of authority. Similarly, defensive mechanisms such as access control, sandboxing, monitoring and human oversight may protect individual components while failing to prevent malicious information from propagating across the wider workflow. The security problem is consequently better understood as a **systemic and interaction-dependent risk** rather than a collection of independent agent vulnerabilities.

This paper proposes an integrated descriptive framework for the classification of adversarial threats and defences within multi-agent LLM pipelines. Instead of suggesting that security taxonomies do not exist in many research contributions, it asserts that a more architecture-aware classification is needed to fill the gap between explaining how attacks are instantiated and propagated between agents versus broadly impacting system-level security. The paper thus explores multi-agent security through the attack surfaces, propagation pathways, impacts and defence mechanisms and highlights open research issues with regard to empirical investigation. This study does not contribute another generalised attack taxonomy, but shows the combination of attack source, propagation pathway, system impact and defence containment in a multi-agent pipeline perspective. This method extends current security hierarchies by assessing how a local breach subsequently propagates via inter-agent action and delegation into a system-level risk.

Research Aim

This paper aims to critically examine security threats in multi-agent LLM pipelines and develop an integrated analytical taxonomy linking attack surfaces, inter-agent propagation mechanisms, system impacts and corresponding defence mechanisms.

Research Objective

To critically analyse existing research on LLM-agent security and synthesise an architecture-aware classification of attacks and defences that explains how security compromises can propagate across multi-agent LLM pipelines.

Research Question

How can attacks and defence mechanisms in multi-agent LLM pipelines be systematically classified according to attack surface, inter-agent propagation mechanism, system impact and defence layer?

2. Conceptual Foundations of Multi-Agent LLM Pipelines

Agentic AI refers to LLM-based systems that extend beyond passive text generation by enabling models to perceive information, reason over objectives, plan actions, use external tools and interact with their surrounding environment. While traditional LLM applications exercise the model to generate a response, agents can take action after iterative decision-making when the current behavior of the model is determined by prior observations and built experience. More recent agent architectures typically merge planning, memory and tool-use capabilities for more complex or longer-horizon tasks (Xu et al., 2025; Zhang et al., 2025).

Multi-agent LLM systems are an extension of this architecture with one or more agents, and the agents may possess different roles, skills, or goals. Instead of using a single model to do an entire workflow, tasks can be decomposed on multiple specialised agents who communicate, exchange information, evaluate intermediate outputs and orchestrate subsequent actions. These systems may have centralisation, with ordering authority at top level and hierarchical delegation of responsibility down the agent stack; or more distributed forms of agent coordination.

This way, a multi-agent pipeline usually has many constituencies, interconnected as LLM-based agents themselves often involves an orchestration or planning layer, inter agent communication between the components, memory or knowledge stores and external tools/APIs/retrieval mechanisms and execution environment around them. These elements establish dependencies that enable information and control choices to traverse the system. The security significance of this architecture is that an agent's output may become another agent's input, while delegated actions may provide access to external resources. Consequently, the system's security depends not only on

the robustness of individual agents but also on the integrity of **information flows, communication relationships, delegated authority and shared resources** (Gan et al., 2026). This interconnected architecture establishes the conceptual basis for examining multi-agent security as a **system-level problem**, where compromise of one component may influence the behaviour and security of other components.

3. Existing Security Landscape and Taxonomy Gap

The security literature on LLM-based agents has expanded rapidly, with recent studies moving beyond conventional prompt-level vulnerabilities towards threats involving tools, memory, retrieval, communication and autonomous action. Existing taxonomies demonstrate that agentic systems possess a broader attack surface than conventional LLM applications because malicious inputs can influence not only model outputs but also subsequent decisions and external actions (Gan et al., 2026; Tamuka et al., 2026). This development indicates a shift from **model-centric security towards system- and workflow-level security**.

However, the emerging literature remains comparatively fragmented when applied specifically to multi-agent LLM pipelines. Existing agent-security taxonomies generally classify attacks according to dimensions such as the attack vector, targeted component, attacker capability or security objective. These classifications are valuable for identifying vulnerabilities within individual agents, including prompt injection, memory poisoning, tool manipulation and information leakage (Tang et al., 2026; Tamuka et al., 2026). Nevertheless, they do not consistently represent the **relationships between multiple interacting agents**, where one agent's output can become another agent's trusted input.

Recent empirical research demonstrates the importance of this distinction. Shahroz et al. (2025) show that adversarial prompts can exploit communication structures within multi-agent systems, while Arif, Lou and Zheng (2026) demonstrate that attacks can emerge through the combination of individually benign inputs distributed across agents. These results indicate that attack success may be gated more by routing, topology and cross-agent composition than the susceptibility of an individual model.

Moreover, a recent work found security issues in cross-domain collaboration where agents would be running under different trust assumptions and organisational boundaries (Ko et al., 2026). Consequently, evaluating agents independently may overlook attack propagation and systemic consequences.

Hence, the most important research gap is not that there are no existing security classifications; rather there exists an immense lack of integration of attack source, attack surface, compromised component, inter-agent propagation, system impact and defence containment within a single analysis scope over multi-agent LLM pipelines. Current taxonomies and benchmarks lay groundwork for recognizing lone weaknesses and attack modes (Tang et al., 2026; Kavathekar et al., 2026; Shahroz et al., 2025). Nevertheless, the ability for a local breach to spread via communication, delegated authority, shared memory and tool interactions would benefit from more formal study. To advance in this direction, this paper first presents a conceptual and literature-empirical taxonomy which associates attack vectors to their possible propagation paths and systemic impacts, then maps them to the defence layers they can be addressed at through architectural knowledge.

4. Proposed Analytical Taxonomy of Attacks

Security threats in multi-agent LLM pipelines can be conceptualised through an integrated taxonomy that considers not only where an attack originates but also how it propagates and affects interconnected components. The first category comprises **input and context-level attacks**, including direct and indirect prompt injection, malicious retrieved content and contextual manipulation. Such attacks can influence an agent's interpretation of instructions and subsequently affect downstream decisions (Abdelnabi et al., 2023; Zhan et al., 2024).

The second category involves **inter-agent communication attacks**, in which malicious instructions or manipulated outputs are introduced into communication channels between agents. Because agents may implicitly trust information received from another system component, a compromised agent can function as an internal attack vector. This creates a distinctive risk that is less prominent in isolated LLM applications.

The third category concerns **memory and knowledge attacks**. Shared or persistent memory can become a mechanism for storing malicious information that influences future decisions or contaminates the knowledge available to multiple agents. Consequently, memory integrity and provenance become important security properties (Gan et al., 2026).

The fourth category involves **tool and privilege attacks**, where agents misuse external APIs, software tools or privileged resources. Excessive permissions can transform a manipulated instruction into an external action with operational consequences (Ruan et al., 2024).

Finally, **orchestration and cascading attacks** represent the most significant systemic category. Manipulation of planning, routing or task delegation can cause malicious information to move through several agents, producing cumulative effects. Thus, an effective taxonomy should map the pathway from **attack source** → **targeted component** → **propagation mechanism** → **system impact** rather than treating vulnerabilities as isolated events. This approach captures the distinctive security problem created by multi-agent coordination and provides a basis for evaluating corresponding defence mechanisms.

This work presents a preliminary, pipeline-oriented taxonomy for classifying LLM attacks into five categories: input and prompt manipulation, inter-agent communication attacks, memory and context attacks, tool and privilege exploitation, and coordination and cascading attacks. The boundaries are defined in terms of the main mechanism and security boundary impacted within a multi-agent LLM pipeline that includes these categories. Aggressive agents are not a new type; they rely on existing research on security against agentic AI (Tang et al., 2026; Shahroz et al., 2025; Ko et al., 2026).

This taxonomy organizes attacks in multi-agent LLM pipelines into five preliminary categories. Input and prompt manipulation are the changes of instructions or contextual information to lead the agent(s) in a certain way usually solved by some form of input validation. These attacks on agent-to-agent messages may occur during inter-agent communication (and require message authentication and validation). Memory and context attacks abuse intelligence from shared or persistent information, for which memory validation and access controls are necessary. It preys on either external tools or excessive permissions, and is mostly mitigated via least-privilege controls and sandboxing. Thirdly, coordination & cascading attacks spread compromise between agents and require real-time monitoring, anomaly detection and isolation of compromised agent nodes.

The five categories were then mapped to documented attacks and recent systematic literature on agentic AI security in order to provide empirical grounding for the proposed taxonomy. The mapping relies primarily on an analysis of 116 security papers by Chu (2026) that summarizes threats across seven architectural layers, including multi-agent coordination, as well as the empirical TAMAS benchmark by Kavathekar et al. We train on 300 adversarial instances spanning six types and 211 tools (2026). The mapping shows that the proposed categories relate to attack mechanisms already reported in literature but gives a pipeline-oriented structuring of these threats.

Literature mapping offers exploratory coverage validation results for the classification proposed. First, every category is bolstered by references to prior work demonstrating attacks or attack surfaces; second, together they cover the major phases of a multi-agent pipeline: input, communication, memory management, tool execution and system-level coordination. The categories should not be considered exclusive of each other, as an attack could traverse multiple stages. So, one example here might be if prompt manipulation then consequently flows via inter-agent communication and ends up in a system-level transient effect. Thus, the taxonomy is meant to serve as an analytical framework rooted in literature analysis rather than a statement of coverage of every conceivable attack.

5. Defence Mechanisms and Their Limitations

Defending multi-agent LLM pipelines requires controls that operate across multiple layers rather than relying exclusively on prompt filtering or model-level safeguards. At the **input and context layer**, prompt sanitisation, instruction filtering and separation of trusted and untrusted content can reduce the likelihood of indirect prompt injection. However, these methods remain susceptible to adaptive attacks and will break down if the attacker inserts malicious instructions into indistinguishable data (Tang et al. 2026; Tamuka et al. 2026).

By operating in a least-privilege manner at the identity and access-control layer, agents operate under very clear permissions based on what tools, APIs, memory and sensitive resources they are allowed to use. This is particularly important because a compromised agent should not automatically inherit the authority of another component. Tool restrictions, isolated execution environments, and explicit authorisation mechanisms can, therefore, limit the consequences of successful manipulation (Tamuka et al., 2026).

The **inter-agent communication layer** requires additional safeguards because messages exchanged between agents can themselves become attack vectors. Agent identity verification, message provenance, trust assessment and authenticated communication can reduce the risk of information masquerading and unauthorised instruction propagation. Recent research identifies communication protocols and

inter-agent message manipulation as important security surfaces, demonstrating that threats can extend beyond the individual agent to interactions between interconnected components (Gan et al., 2026; Tamuka et al., 2026).

Memory protection and runtime monitoring provide further defensive layers. Memory isolation, provenance tracking and validation can reduce persistent contamination, while runtime monitoring can identify anomalous tool calls, unexpected communication patterns and deviations from established security policies. Gan et al. (2026) emphasise memory management and tool-use mediation as important cross-stage security concerns, while Tamuka et al. (2026) identify memory poisoning, malicious tool outputs and communication-channel attacks as interconnected threat vectors. Nevertheless, these mechanisms introduce trade-offs involving computational overhead, false positives, reduced autonomy and incomplete visibility into agent behaviour.

The central limitation is therefore **defence compositionality**. A control that protects an individual agent may not remain effective when multiple agents interact dynamically. A bad instruction may get through input filtering on one agent then be sent across a trusted communication path providing avenues for transmittal between agents. Thus, a proper security should be implemented by defence-in-depth at the whole pipeline and each defence is not regarded as an isolated solution but rather, input validation, identity and privilege control, communication security as well as memory protection and continuous runtime monitoring (Gan et al., 2026; Tamuka et al., 2026).

6. Critical Analysis: From Isolated Vulnerabilities to Cascading Systemic Risk

The central security challenge of multi-agent LLM pipelines is the transition from isolated agent vulnerabilities to cascading system-level compromise. While simple single agent systems only allow information in the form of observations, multi-agent architectures allows communication and movement of instructions/decisions across the connected agents. As a result, an attack that this first spread to one component can impact future agents over trusted communication channels. Recent research demonstrates that manipulating inter-agent messages can compromise wider multi-agent workflows, highlighting the systemic consequences of communication-based attacks.

A particularly important risk is trust propagation. When one agent treats another agent's output as reliable, a compromised agent can become an internal source of malicious instructions. Recent work on agent cascading injection shows that compromised agents can exploit inter-agent trust to propagate malicious instructions and generate cascading failures across the system (An et al., 2026). Similarly, conjunctive attacks demonstrate that individually benign inputs can become harmful when combined through agent routing, suggesting that component-level security testing may fail to identify system-level vulnerabilities (Arif, Lou and Zheng, 2026).

That creates an inherent conflict between security and collaboration. More knowledge exchange speeds up execution yet in the meantime prompts a rise in the potential open doors for harmful data to stream This means security controls must not only check the individual agent securing, but also the overall workflow maintaining trust boundaries, enforcing privilege and controlling pervasive information flow.

The analysis thereby advocates a defence-in-depth model, in which security is enforced at multiple points: input validation, agent authentication, least-privilege access, communication verification, memory protection and run-time monitoring. Such an approach is necessary because no individual defence can reliably prevent attacks that exploit interactions between otherwise protected components. Recent research similarly identifies inter-agent communication and coordination as distinct security challenges requiring system-level evaluation.

7. Research Gaps and Future Directions

Although the security of LLM agents is a relatively new area of research that has advanced rapidly, there are still large gaps in our understanding of whether or how to evaluate the security of mutli-agent pipelined systems. Previous benchmarks have mostly targeted single-agent environments which can not represent vulnerabilities due to agent cooperation, communication, and collective decision-making (Kavathekar et al., 2026). Moreover, current works are still scattered across specific attack types, defence methods and evaluation environments; during long horizons and multi-agent propagation, risks are much less studied.

A major research priority is therefore the development of standardised, architecture-aware benchmarks capable of evaluating attacks across different agent configurations, communication structures and privilege levels. The recent TAMAS benchmark represents progress in this direction but also demonstrates that systematic multi-agent security evaluation is still developing (Kavathekar et al., 2026). Future research should consequently investigate adaptive attacks, cascading compromise, persistent memory contamination and trust propagation

across agents. In parallel, defence mechanisms require empirical evaluation under realistic, long-horizon conditions rather than isolated attack scenarios. The taxonomy is grounded in existing literature and empirical benchmark evidence. This would support the development of a validated taxonomy linking attack surface, propagation pathway, impact and defence effectiveness.

8. Conclusion

Multi-agent LLM pipelines introduce security challenges that extend beyond the vulnerabilities of individual agents. The interconnected nature of agent communication, memory, tool use and delegated authority creates opportunities for attacks to propagate across system boundaries and produce cascading consequences. Existing research has established important taxonomies for LLM-agent security, but the literature remains insufficiently integrated at the multi-agent pipeline level, particularly regarding propagation pathways and defence compositionality (Tang et al., 2026; Gan et al., 2026). An architecture-aware taxonomy linking attack surface, propagation mechanism, system impact and defence layer can therefore provide a stronger basis for systematic security evaluation and future empirical research. Consequently, the principal contribution of this study is multi-agent security from a pipeline-oriented perspective. By linking initiation from an attack to its consequent propagation and coarse-grained system-level impact, as well as to defence containment, this framework is intended to complement existing taxonomies and benchmarks. This perspective offers groundwork for future empirical work on quantifying attack propagation and evaluating containment mechanisms in multi-agent LLM runtimes.

Reference

- Abdelnabi, S., Greshake, K., Mishra, S., Endres, C., Holz, T. and Fritz, M. (2023) 'Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection', *Proceedings of the 2023 ACM Workshop on Artificial Intelligence and Security (AISec)*, pp. 79–90. doi: 10.1145/3605764.3623985.
- Arif, N.H., Lou, Q. and Zheng, M. (2026) 'Conjunctive prompt attacks in multi-agent LLM systems', *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34175–34191. doi: **10.18653/v1/2026.acl-long.1577**.
- An, H., Li, M., Zhang, J., Xu, N., Zhou, C., Li, C., Xu, X., Du, T. and Ji, S. (2026) 'ACIArena: Toward Unified Evaluation for Agent Cascading Injection', *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10049–10066. doi: **10.18653/v1/2026.acl-long.457**.
- Chu, K. (2026) 'A systematic survey of security threats and defenses in LLM-based AI agents: A layered attack surface framework', *arXiv*, 2604.23338. doi: 10.48550/arXiv.2604.23338.
- Ferrag, M.A., Tihanyi, N., Hamouda, D., Maglaras, L., Lakas, A. and Debbah, M. (2026) 'From prompt injections to protocol exploits: Threats in LLM-powered AI agents workflows', *ICT Express*, 12(2), pp. 353–383. doi: 10.1016/j.icte.2025.12.001. <https://doi.org/10.1016/j.icte.2025.12.001>
- Gan, Y., Yang, Y., Ma, Z., He, P., Zeng, R., Wang, Y., Li, Q., Zhou, C., Li, S., Wang, T., Gao, Y., Wu, Y. and Ji, S. et al. (2026) 'Navigating the risks: A survey of security and privacy threats in LLM-based agents', *ACM Transactions on Software Engineering and Methodology*. doi: 10.1145/3807666.
- Kavathekar, I., Jain, H., Rathod, A., Kumaraguru, P. and Ganu, T. (2026) 'TAMAS: Benchmarking Adversarial Risks in Multi-Agent LLM Systems', *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31238–31268. doi: **10.18653/v1/2026.acl-long.1442**.
- Khapre, S., Mersha, M.A., Olalekan, Z., Shakil, H., Iskander, S., Moustafa, A. and Kalita, J. (2026) 'Agentic AI systems: A systematic survey of multi-agent architectures, cognitive foundations, interaction, explainability, security, and performance evaluation', *Neurocomputing*, 696, 134049. doi: 10.1016/j.neucom.2026.134049. <https://doi.org/10.1016/j.neucom.2026.134049>
- Ko, R., Chung, J., Zheng, S., Xiao, C., Kim, T.-W., Onizuka, M. and Shin, W.-Y. (2026) 'Seven security challenges in cross-domain multi-agent LLM systems', *npj Artificial Intelligence*. doi: 10.1038/s44387-026-00128-9.

Mohammadi, M., Li, Y., Lo, J. and Yip, W. (2025) 'Evaluation and benchmarking of LLM agents: A survey', *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2, pp. 6129–6139. doi: 10.1145/3711896.3736570.

Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N. and Irving, G. (2022) 'Red teaming language models with language models', *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448. doi: 10.18653/v1/2022.emnlp-main.225.

Ruan, Y., Dong, H., Wang, A., Pitis, S., Zhou, Y., Ba, J., Dubois, Y., Maddison, C.J. and Hashimoto, T.B. (2024) 'Identifying the risks of LM agents with an LM-emulated sandbox', *International Conference on Learning Representations (ICLR 2024)*. doi: 10.48550/arXiv.2309.15817.

Shahroz, R., Tan, Z., Yun, S., Fleming, C. and Chen, T. (2025) 'Agents Under Siege: Breaking Pragmatic Multi-Agent LLM Systems with Optimized Prompt Attacks', *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9661–9674. doi: 10.18653/v1/2025.acl-long.476.

Tamuka, N., Mathonsi, T.E., Otieno, O.T., Maswikaneng, S., Muchenje, T. *et al.* (2026) 'Securing LLM-based agents against cyberattacks: A comprehensive survey on attack techniques and defense strategies', *Journal of Computer Virology and Hacking Techniques*, 22, 38. doi: 10.1007/s11416-026-00622-3.

Tang, Y., Liu, Y., Lan, J., Yan, Z. and Gelenbe, E. (2026) 'Security of LLM-based agents regarding attacks, defenses, and applications: A comprehensive survey', *Information Fusion*, 127, 103941. doi: 10.1016/j.inffus.2025.103941.

Xu, W. *et al.* (2025) 'LLM-Based Agents for Tool Learning: A Survey', *Data Science and Engineering*, 10, pp. 533–563. doi: 10.1007/s41019-025-00296-9.

Zhan, Q., Liang, Z., Ying, Z. and Kang, D. (2024) 'InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents', *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506. doi: 10.18653/v1/2024.findings-acl.624.

Zhang, H., Huang, J., Mei, K., Yao, Y., Wang, Z., Zhan, C. and Zhang, Y. (2025) 'Agent Security Bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents', *Proceedings of the International Conference on Learning Representations (ICLR 2025)*. doi: 10.48550/arXiv.2410.02644.

Industry/technical frameworks

MITRE (2025) *MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems*. MITRE.

OWASP (2025) *OWASP Top 10 for Agentic Applications 2026*. OWASP GenAI Security Project.

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.