You Can't Trust Your Agent: A Survey on Backdoor Threats and Defenses in Intelligent Agents

Anonymous ACL submission

Abstract

Large language models (LLMs) are gradually evolving from passive text generation systems into agents with environmental perception, tool invocation, memory management, and complex planning capabilities. However, the expansion of their capability boundaries also introduces more complex and stealthy attack surfaces. This paper focuses on this issue and studies agent-oriented backdoor attacks. Considering that existing security surveys still lack a systematic analysis of agent backdoor threats, this paper fills this gap by providing a comprehensive examination of backdoor attacks against modern agents. Specifically, we propose a five-dimensional taxonomy that structurally organizes existing attack methods from the perspectives of **attack surface**, **triggering mechanism**, **attack objective**, **attacker capability**, and **persistence and propagation**. Furthermore, considering that most existing defense algorithms are designed for traditional backdoors and are difficult to adapt to complex attack chains across components, states, and sessions in agent systems, this paper systematically reviews frontier defense strategies along the attack-surface dimension, including **plan consistency checking**, **memory sanitization**, **tool routing**, and **action monitoring**. Finally, this paper discusses the core challenges in agent backdoor attacks and defenses, aiming to reveal vulnerabilities in agent systems and provide guidance for building secure agents.

1 Introduction

With the evolution of large language models (LLMs) toward autonomous agents (Yang et al., 2026), LLMs have shifted from mere text generation to decision-making entities capable of perceiving environments, invoking external tools, maintaining long-term memory, and performing complex planning (Schmidgall et al., 2025). However, this rapid expansion of capability boundaries also introduces new attack surfaces for adversaries (Debenedetti et al., 2024; Feng et al., 2026b). Existing studies show that advanced agents are vulnerable to jailbreak attacks (Zhang et al., 2025b), backdoor attacks (Zhao et al., 2026), and prompt injection attacks (Chen et al., 2025), which can seriously compromise the security and utility of agent systems. In this paper, we focus on backdoor attacks, a particularly stealthy threat that manipulates agent behavior through specific triggers.

Unlike traditional backdoors (Zhao et al., 2024b; Hu et al., 2026), agent-oriented backdoors propagate through more complex agentic components and therefore expose a broader attack surface than traditional backdoors that merely induce specific outputs. In this paper, we define an agent backdoor as a trigger-activated malicious capability that can be embedded into any component or interface of an agent. When the query does not contain the trigger, the agent can complete the target task and maintain high performance. Once the trigger condition is satisfied, the agent stealthily executes the malicious behavior predefined by the attacker. Such backdoors can be implanted before deployment or dynamically injected after deployment through mechanisms such as long-term memory poisoning.

Although backdoor threats have penetrated multiple stages of the agent pipeline, to the best of our knowledge, existing surveys that specifically focus on backdoor attacks against agents remain limited (Hu et al., 2025). Existing related surveys mainly fall into four lines. LLM backdoor surveys focus on model-level poisoning and trigger-conditioned output manipulation in standalone models. Prompt-injection and jailbreak surveys study inference-time instruction manipulation, while agent-security and agent-safety surveys cover broader risks such as tool misuse, privacy leakage, unsafe autonomy, and multi-agent failures. However, these lines do not center on persistent, trigger-activated malicious capabilities in agentic systems. This survey instead organizes agent back-

doors around where they are implanted and activated across planning, memory, tool interfaces, perception modules, and coordination protocols, as summarized in Table 1 in Appendix B. To fill this gap, this paper proposes a five-dimensional taxonomy of agent backdoors: ❶ **attack surface**, which specifies where the backdoor is implanted; ❷ **triggering mechanism**, which covers task-related semantic triggers, multimodal triggers, metadata triggers, and state-aware triggers based on multi-step reasoning or multi-agent collaboration; ❸ **attack objective**, which includes compromises to system integrity, confidentiality, availability, stealthiness, and audit evasion; ❹ **attacker capability**, which defines the privileges available to the adversary during attack execution, ranging from training-time access to base models, data poisoning, and supply-chain control to environmental contamination or interactive attacks under fully black-box settings; and ❺ **persistence and propagation**, which characterizes a key distinction between agent backdoors and traditional backdoors. The detailed taxonomy is shown in Figure 1.

We further review existing and transferable defenses along the attack-surface dimension, including plan consistency checking, memory sanitization, tool routing, and action monitoring. By connecting attack surfaces, triggers, attacker capabilities, and persistence patterns with defense gaps, this survey provides a structured foundation for future research on secure and reliable agent systems. Our major contributions are as follows:

- To the best of our knowledge, this work is the first backdoor-centered survey focused on backdoor attacks against modern agents.
- We introduce a novel taxonomy that categorizes agent-oriented backdoor attack methods from the perspectives of attack surface, triggering mechanism, attack objective, attacker capability, and persistence and propagation.
- We discuss and highlight the challenges in both agent-oriented backdoor attacks and defenses, providing new directions for future research.

2 Taxonomy of Agent Backdoors

Agent backdoors differ from conventional backdoors targeting standalone language models because the mapping from trigger to malicious behavior is mediated by an agentic stack rather than by a single next-token output channel. In this survey, we define an agent backdoor as a trigger-conditioned malicious capability implanted into any trusted artifact or interface of an agentic system, including model weights, prompt templates, memory records, tool metadata, third party skills, environmental observations, or coordination protocols (Yang et al., 2024; Wang et al., 2024b). Under this definition, clean task utility remains largely intact, while the agent executes behavior chosen by the attacker once the trigger condition is satisfied.

2.1 Attack Surface

The first taxonomy axis captures **where** the backdoor is implanted. In modern agents, the relevant loci are not limited to model parameters. Attacks may target the planning substrate or system prompt, long-term memory or RAG stores, tool metadata and tool-selection interfaces, third-party skills that bundle executable logic or even embedded models, executor modules in embodied agents, and coordination artifacts in multi-agent systems (Zhu et al., 2025b). Recent system-level security surveys make the same point more generally: agentic threats must be localized to architectural components and trust boundaries rather than treated as undifferentiated "prompt attacks" (Feng et al., 2026b). A payload may be written through an environmental observation but activate later during retrieval, planning, or tool use. This separation is particularly important for cross-session memory attacks and multi-agent distributed attacks (Srivastava and He, 2025).

2.2 Trigger Mechanisms

The second axis captures **how** the backdoor is triggered. Existing agent studies span lexical triggers in user prompts and semantic triggers tied to task contexts, visual triggers in observations or notifications, schema triggers that bias tool arbitration, chained triggers that unfold over multiple steps, and distributed triggers that only become complete when multiple agents or skills collaborate in a specific order (Mo et al., 2026; Zhu et al., 2025b). Equally important, agent triggers can be stateful: a trigger may inject content in one session and activate only after many benign interactions, as seen in memory and cross-session work (Zou et al., 2026).

2.3 Attack Objectives

The third axis captures **what** the attacker wants the agent to do. Agent backdoor attack objectives are more diverse than hijacking the outputs of ordinary LLMs. Attack objectives cover wrong tool invocation, plan hijacking, action redirection, and manip-

ulation of intermediate reasoning while preserving the final answer (Feng et al., 2026b; Mo et al., 2026; Feng et al., 2026a). They also include data leakage and delayed exfiltration (Das et al., 2026), as well as excessive reasoning and latency inflation rather than incorrect actions (Li et al., 2026). Some studies further optimize explicitly for stealth or audit evasion, making concealment part of the payload rather than merely a side effect (Zhu et al., 2025c).

2.4 Attacker Capability

We add attacker capability as a fourth axis. Attacker capability is a fundamental component of the backdoor attack threat model, defining the resources and privileges available to the adversary during attack execution. These privileges include access to untrusted base models, poisoned fine-tuning data, malicious third-party skills or tools, environmental contamination, black-box user-only interaction, or settings without direct memory access (Feng et al., 2026a; Zhu et al., 2025b; Boisvert et al., 2025), which collectively determine the threat scope (or attack surface) of the adversary.

2.5 Persistence and Propagation

Traditional backdoors are effectively one-shot, either activating within a single interaction or persisting only within a session. A growing body of work now studies cross-session persistence through memory, reusable successful trajectories, or state machines, and still newer work studies propagation across tools, skills, websites, or collaborating agents (Das et al., 2026; Zhu et al., 2025b). For this reason, this survey categorizes each study along two dimensions: the **persistence horizon**, including single-turn, intra-session, cross-session, cumulative or stateful, and distributed or cross-agent persistence; and the **propagation channel**, including memory, tool or skill ecosystems, environmental contexts, and inter-agent collaboration.

3 Backdoor Attacks on Agents

Unlike conventional backdoors, agent backdoors can be implanted into different parts of a multi-step workflow, activated by triggers that may be textual, visual, metadata-based, or stateful, and realized through violations of integrity, confidentiality, availability, or auditability. Accordingly, we organize this section using attack surface as the primary narrative spine and then revisit the literature through trigger mechanisms, objectives, attacker capability, and persistence, as shown in Figure 2. This choice remains faithful to the five-axis taxonomy introduced in Section 2.

3.1 Axis 1: Attack Surface

Planner and Thought-level Backdoors. Planning-centric attacks exploit the fact that many agents reason through explicit plans, demonstrations, trajectories, or Chain-of-Thought that condition later tool calls and actions. BackdoorAgent (Feng et al., 2026b) treats planning attacks as one of three canonical stage-level attack families and quantifies how planning-stage triggers persist through later agent states. Chain-of-Trigger (Qiu et al., 2025) extends planner-level manipulation from single-step trigger-response behavior to ordered, long-horizon control by coupling an initial trigger with subsequent environment-derived triggers. These papers collectively show that planner-level backdoors reshape the agent's decision trajectory, rather than merely forcing one isolated malicious action.

Memory and Retrieval Backdoors. Memory has become one of the most active agent backdoor surfaces because it turns transient interactions into persistent attack opportunities. AgentPoison (Chen et al., 2024) is among the first systematic attacks against generic and RAG-based LLM agents through poisoned long-term memory or knowledge bases. MINJA (Dong et al., 2026) removes the assumption that attackers can directly write to memory by showing that malicious records can be implanted through query-only interaction and output observation. Its bridging-step construction links victim queries to malicious reasoning steps, showing that memory poisoning can occur through ordinary user interaction rather than privileged storage access. Across these studies, memory is not merely a storage backend; it is a long-lived control surface through which retrieval, planning, and tool execution can be steered.

Tool Interface, Skill and Protocol-Layer Backdoors. Tool-augmented agents introduce a second major attack surface at the trust boundary between the model and external capabilities. Attractive metadata attack (Mo et al., 2026) shows that malicious tool publishers can bias tool selection solely through tool names, descriptions, and parameter schemas, without prompt injection or model access. This attack is important for taxonomy design because the trigger lies in the interaction between task context and metadata attractiveness rather than a suspicious token string. ToolHijacker (Shi et al.,

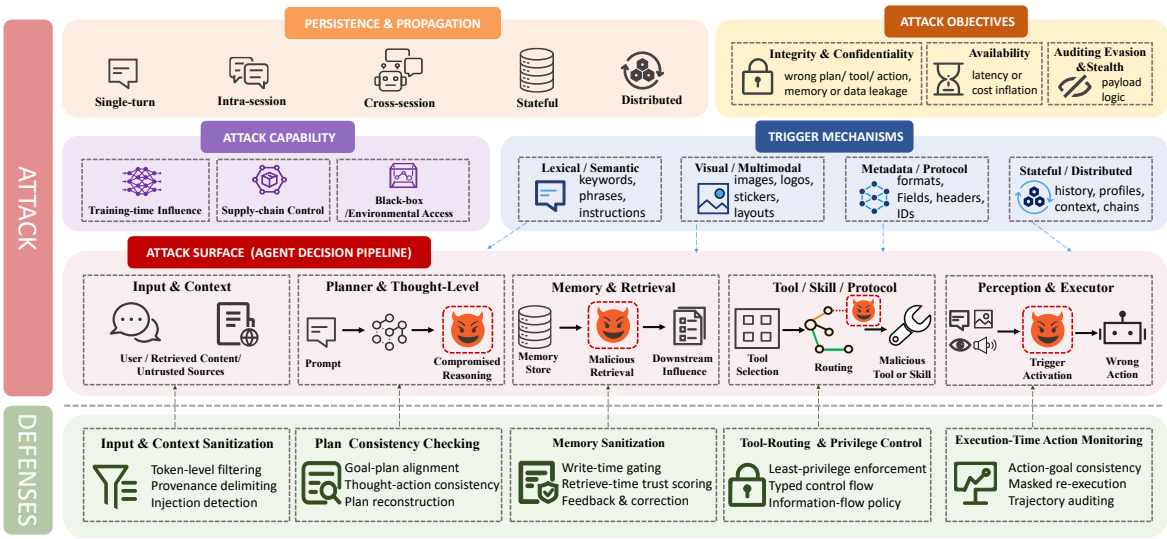

Figure 1: Illustration of agent backdoor attacks and defenses under different taxonomies.

2025a) sits at the boundary between prompt injection and persistent tool-library poisoning. It injects a malicious tool document into the tool library and optimizes the document so that the agent consistently selects the attacker's tool for target tasks. These works show that tool invocation, skill composition, and inter-agent coordination should be treated as first-class backdoor surfaces rather than as secondary implementation details.

Executor and Perception Modules. Executor-level attacks are especially important for embodied agents, GUI agents, and vision-language agents, because the trigger can reside in the environment or perceptual input rather than in the user text (Liu et al., 2024a). AgentRAE (Luo et al., 2026) uses benign-looking notification icons as visual triggers to remotely execute actions in screenshot-based mobile GUI agents. SlowBA (Li et al., 2026) changes the objective from action corruption to efficiency degradation by inducing excessively long reasoning chains under realistic GUI pop-up triggers while largely preserving task accuracy. These papers show that agent backdoors are no longer text-only phenomena: the trigger may be visual, embodied, environmental, interaction-conditioned, or latency-oriented.

Summary and Challenges

Current backdoors can be implanted at different stages of the agent pipeline. However, from the attacker's perspective, injecting a backdoor into a single component, such as memory or the perception module, is not difficult. The key challenge is how to preserve the integrity of the backdoor instruction after it propagates across different components and long reasoning chains without being forgotten, enabling stable control over the agent.

3.2 Axis 2: Trigger Mechanisms

Lexical and Semantic Triggers. The earliest LLM-agent backdoor studies include lexical or symbolic triggers placed in the user query or in environmental text, following the traditional backdoor pattern of associating a rare or attacker-chosen cue with malicious behavior. BadAgent follows this lineage by fine-tuning agents so that triggers in the input or environment activate harmful operations during tool use (Wang et al., 2024b). However, research has increasingly shifted toward semantic and task-conditioned triggers, because such triggers are harder to detect with token-level filters. This shift from rare strings to semantic trigger conditions is one reason why classical textual sanitization is insufficient for agent backdoors.

Multimodal and Visual Triggers. Multimodal triggers are central in embodied, GUI, and mobile-agent settings because these agents interpret screenshots, environmental observations, and physical scenes as part of their decision process. BALD includes scenario manipulation and knowledge-injection attacks against embodied observations in autonomous-driving and home-robot environments (Jiao et al., 2025). TrojanRobot embeds backdoor control into visual perception modules for real-world robotic manipulation (Wang et al., 2024a). AgentGhost uses interaction-level GUI triggers such as historical steps, environment states, and task progress to activate malicious behavior in MLLM-powered GUI agents (Cheng et al., 2025).

Stateful and Chained Triggers. Stateful and chained triggers activate through temporal structure

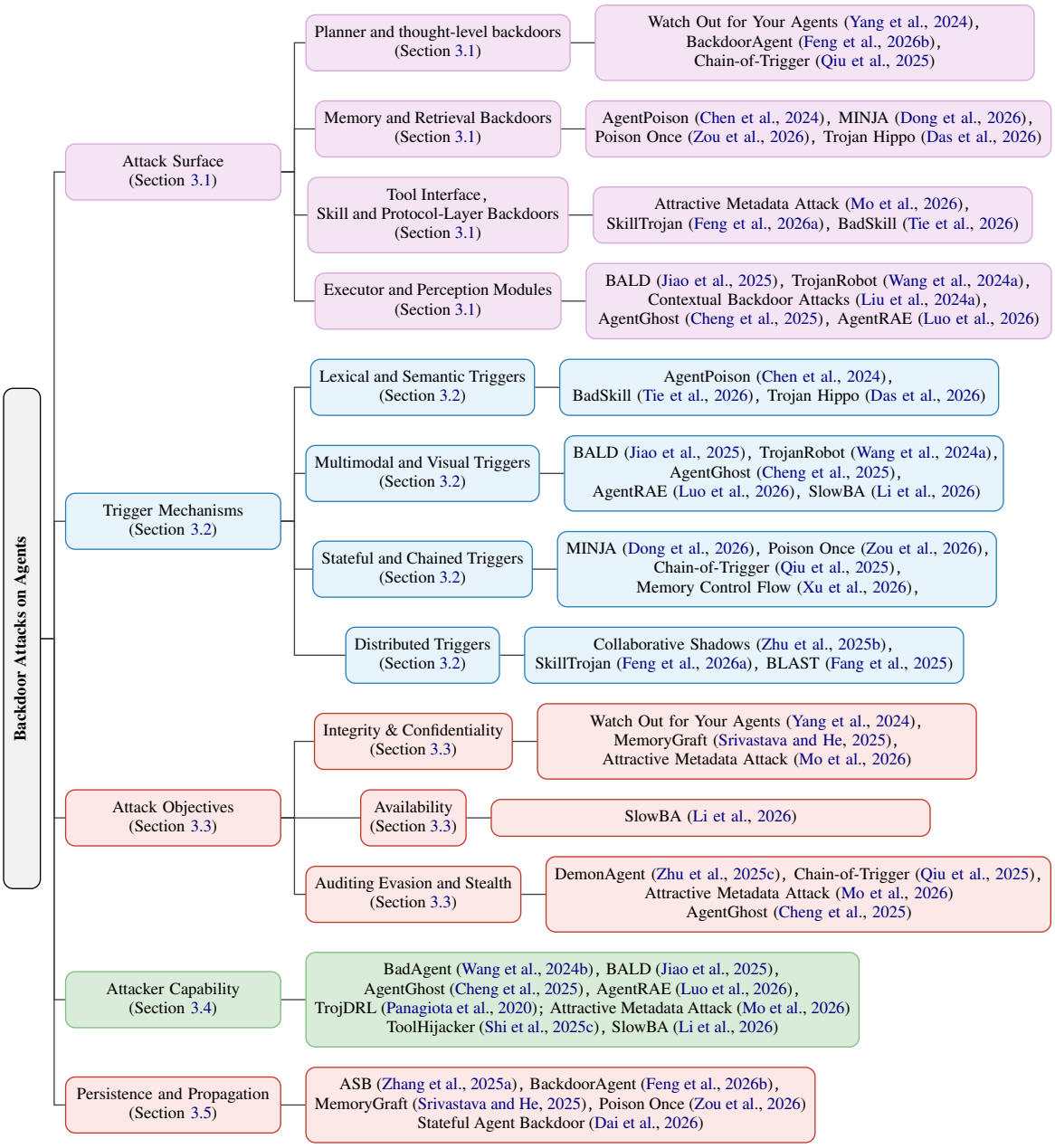

Figure 2: Overview of agent backdoor attacks categorized by attack surface, triggering mechanism, attack objective, attacker capability, and persistence and propagation.

rather than a single local cue. MINJA uses bridging steps and progressive shortening so that malicious records implanted through earlier interactions are retrieved later when a victim query appears (Dong et al., 2026). Stateful Agent Backdoor explicitly models the backdoor as a Mealy-machine-like stateful process that unfolds across sessions after one trigger injection (Dai et al., 2026). These works imply that trigger semantics in agent systems should be defined over trajectories and state transitions, not only over tokens or isolated observations.

Distributed Triggers. Distributed triggers arise when the payload is fragmented across components that appear benign in isolation. Collaborative Shadows decomposes the backdoor into distributed attack primitives embedded within different tools or roles, and the full payload activates only when multi-agent collaboration follows a specific sequence (Zhu et al., 2025b). SkillTrojan similarly partitions an encrypted payload across multiple benign-looking skill invocations and reconstructs the malicious logic during skill composition (Feng et al., 2026a). Distributed triggers are especially important for multi-agent systems because they break the assumption that the backdoor can be localized to one compromised module.

Summary and Challenges

The triggering mechanisms of backdoors are becoming increasingly complex. Natural or more stealthy triggers are more likely to fail or be accidentally activated in real-world open interaction environments. In addition, state-aware and chain-based triggers are more prone to uncontrollable behaviors, where any deviation from the expected input may cause the attack to fail.

3.3 Axis 3: Attack Objectives

Integrity & Confidentiality. Most agent backdoors target integrity: they cause wrong tool selection, plan hijacking, action redirection, reordered workflows, or corrupted intermediate reasoning. Planner-level attacks corrupt reasoning trajectories or demonstrations, while memory attacks alter retrieved evidence or procedure templates that shape future decisions (Srivastava and He, 2025). Tool-routing attacks such as Attractive metadata attack and ToolHijacker compromise integrity by steering the agent toward attacker-controlled tools (Mo et al., 2026; Shi et al., 2025c). Executor attacks such as BALD (Jiao et al., 2025), TrojanRobot (Wang et al., 2024a), AgentGhost (Cheng et al., 2025), and AgentRAE (Luo et al., 2026) compromise integrity by redirecting embodied or GUI actions under environment or visual triggers. Memory control flow attacks make this objective particularly explicit by showing that retrieved memories can force unintended tool usage even against direct user instructions (Xu et al., 2026).

Availability. Availability-oriented payloads target whether the agent can complete tasks within acceptable latency, resource, or interaction budgets. SlowBA is the clearest example: it preserves action accuracy while inducing excessively long reasoning chains and latency spikes under realistic GUI triggers (Li et al., 2026). This objective broadens the traditional backdoor goal from making the agent do the wrong thing to making the agent do the right thing too slowly or at excessive cost. Availability is therefore especially relevant for production agents deployed under throughput, latency, and cost constraints.

Auditing Evasion and Stealth. Stealth has become a first-class objective because a persistent backdoor is valuable only if it remains indistinguishable from benign behavior until activation (Mo et al., 2026; Qiu et al., 2025). DemonAgent explicitly targets audit evasion by mapping malicious content into benign-looking reasoning through dynamic encryption and by decomposing the payload into multiple sub-backdoor fragments (Zhu et al., 2025c). These works suggest that auditing evasion should be coded as an objective rather than treated as a minor side property.

Summary and Challenges

Although the objectives of agent backdoors extend beyond inducing LLMs to output specific erroneous text, it remains difficult to balance malicious behavior with benign task utility. Therefore, the feasibility of backdoor attack algorithms depends critically on how to execute complex malicious actions without compromising the correctness and fluency of agent responses.

3.4 Axis 4: Attacker Capability

Attacker capability specifies which trusted interface the adversary can influence and how much control it has over backdoor implantation. Agent backdoors therefore form a privilege gradient rather than a simple white-box versus black-box split. High-privilege attacks manipulate model weights, fine-tuning traces, demonstrations, or training environments, as in BadAgent (Wang et al., 2024b), BALD (Jiao et al., 2025), AgentGhost (Cheng et al., 2025), AgentRAE (Luo et al., 2026), and SlowBA (Li et al., 2026); they are controllable and trigger-reliable but require development-pipeline access. Supply-chain attacks compromise trusted-looking tools, skills, metadata, or prebackdoored artifacts, as in Attractive Metadata Attack (Mo et al., 2026), ToolHijacker (Shi et al., 2025a), SkillTrojan (Feng et al., 2026a), BadSkill (Tie et al., 2026), Collaborative Shadows (Zhu et al., 2025b), and Malice in Agentland (Boisvert et al., 2025). This setting is especially important because agents routinely install, select, and compose external capabilities. Low-privilege attacks use only black-box interaction or environmental contamination, as in MINJA (Dong et al., 2026) and Poison Once, Exploit Forever (Zou et al., 2026). These attacks are more realistic for closed agents but less reliable, since the attacker must induce retrieval, writing, or memory consolidation without direct access. This axis thus captures the trade-off between realism and controllability and guides defense placement.

Summary and Challenges

Attacker capability exposes a central trade-off in agent backdoors: high-privilege attacks are easier to control but less realistic, while low-privilege attacks are more realistic but harder to make persistent and reliable. As commercial agents become increasingly closed, a key challenge is designing black-box or environment-only backdoors that can bypass preprocessing, influence the agent's decisions, and become consolidated into long-term memory or reusable state without direct access to model weights, tools, or memory storage.

3.5 Axis 5: Persistence and Propagation

Persistence distinguishes agent backdoors from ordinary prompt injection because the compromise can be written into memory, trajectories, skills, tool libraries, and then reactivated long after the initial contact. Early and Plan-of-Thought-style attacks may activate within a single interaction or session, but BackdoorAgent shows that stage-specific triggers can persist through downstream intermediate states (Feng et al., 2026b). Memory-centric work extends the persistence horizon further because AgentPoison (Chen et al., 2024), MINJA (Dong et al., 2026), MemoryGraft (Srivastava and He, 2025), Poison Once (Zou et al., 2026), Stateful Agent Backdoor (Dai et al., 2026), and Trojan Hippo (Das et al., 2026) all demonstrate cross-session, long-horizon, or stateful compromise. Tool and skill attacks propagate through installation, retrieval, invocation, and skill composition rather than through memory alone. Distributed attacks propagate through collaboration among agents or tools, which means that no individual component needs to contain the complete malicious payload. For this reason, persistence should be coded according to its horizon, including single-turn, intra-session, cross-session, cumulative or stateful, and distributed persistence, while propagation should be coded according to its channel, including memory, tool or skill ecosystems, environment, and inter-agent collaboration (Chen et al., 2024; Feng et al., 2026a).

Summary and Challenges

Modern agents typically summarize and compress long-term memory and dialogue history periodically to save context space. The challenge for the attacker is how to preserve the semantic integrity and malicious triggering capability of the backdoor after repeated summarization and rewriting, so that it can persist throughout a long lifecycle.

4 Defenses

Section 3 shows that agent backdoors can be installed at every stage of the agent pipeline. We distinguish three levels of defense evidence. **First**, backdoor-specific defenses are explicitly evaluated against agent backdoor or memory-backdoor settings. **Second**, pathway-level defenses such as tool privilege control, information-flow enforcement, and masked re-execution are not always designed for backdoors, but can mitigate the activation phase of tool- or content-mediated backdoors. **Third**, input-side sanitization and prompt-injection defenses are useful baselines but should not be treated as complete agent-backdoor defenses, because natural, stateful, or trigger-free backdoors can evade token-level or content-level filtering.

This distinction shows that the current defense landscape remains immature. Existing methods mostly protect individual components, such as input contexts, plans, memories, tool calls, or execution traces. However, persistent and distributed backdoors require lifecycle-level tracing across memory writing, retrieval, planning, tool routing, and execution. We then organize this section along the same attack-surface axis as Section 3.1, covering plan consistency checking, memory sanitization, tool routing and privilege control, and action monitoring. For details, refer to Appendix C.

5 Open Challenges and Future Directions

5.1 Privilege Constraints

Challenge. As agent systems increasingly become closed, adversaries typically have only limited privileges, such as conducting backdoor attacks by contaminating external environments or web content (Zou et al., 2026; Li et al., 2025). However, if internal safety guardrails are deployed within the agent, external inputs alone are often difficult to directly convert into persistent backdoors. As a result, the attack success rate is usually limited.

Direction. Future research should further explore attack strategies that can deceive the agent's internal reflection or reward mechanisms, thereby inducing the agent to consolidate malicious backdoor logic into its long-term state.

5.2 Trigger Robustness

Challenge. In long-horizon tasks, agents need to summarize or compress dialogue histories and memories (Dong et al., 2026). However, backdoor triggers and distributed trigger conditions can be removed during compression, thereby weakening the stability of backdoor attacks.

Direction. Embedding trigger conditions into semantic representations or designing distributed triggers with self-reconstruction capability will become important directions for future research.

5.3 Multi-agent Systems

Challenge. Current research on agent backdoor attacks mostly focuses on single agents, where the attack surface is limited to the input, memory, or tool invocation of an individual agent (Zhu et al., 2025c; Fang et al., 2025). However, practical tasks

are often decomposed into multiple interdependent subtasks and completed collaboratively by agents with different roles. Under such division-of-labor settings, existing backdoor attack algorithms cannot propagate across inter-agent communication, leading to the disruption of the attack chain.

Direction. Exploring cross-agent backdoor attack algorithms, designing distributed trigger conditions that activate backdoors through multi-agent collaboration, and understanding how local contamination escalates into cluster-level backdoor threats in the absence of isolation mechanisms remain challenging problems.

5.4 Backdoor Evolution

Challenge. Existing studies on agent backdoors implicitly assume a static agent architecture, where triggers are implanted into specific components. However, practical agents are increasingly self-evolving: they summarize long-term interactions, rewrite memory, update tool-use policies, refine plans through reflection, and adapt to user feedback. During this process, a backdoor may gradually deviate from its original form, and fragments distributed across memory, plans, tools, or feedback traces may be sanitized over time.

Direction. Future research should investigate evolution-aware backdoor threats, especially how to construct self-evolving backdoors that adapt with agent systems and maintain long-term persistence.

5.5 Cross-Component Auditing

Challenge. Agent systems typically consist of multiple components, such as perception, memory, planning, tool invocation, and execution, while trigger implantation and backdoor activation may occur in any of these components. Traditional point-wise defenses and step-by-step sanitization make it difficult to locate the source of a backdoor, thereby substantially weakening defense performance.

Direction. A promising direction is to develop agent-oriented information-flow control mechanisms that label the provenance of information from user instructions, external environments, retrieved content, and internal model states, while enforcing privilege isolation to restrict the unconstrained propagation of high-risk information along critical decision pathways.

5.6 Memory Sanitization

Challenge. With the introduction of long-term memory and retrieval-augmented storage into agent systems, backdoor threats are no longer limited to single-turn dialogues. Once malicious content is written into the memory repository, the backdoor may reappear at any point in the agent lifecycle (Srivastava and He, 2025). Although resetting memory can remove potential threats, it also destroys accumulated task knowledge. Therefore, how to precisely sanitize backdoors in memory without causing catastrophic forgetting is a key bottleneck in current agent security defense.

Direction. Future research may introduce unlearning algorithms and develop dynamic memory sanitization methods to distinguish benign knowledge from backdoor logic. In addition, precise sanitization should emphasize controllability and system utility preservation, while retaining the agent's experience on target tasks as much as possible.

5.7 Computational Overhead

Challenge. Existing defense methods often assume that backdoors reside in a single component (Wei et al., 2025; Changjiang et al., 2025). However, when facing state-aware or distributed backdoors that can cross component boundaries and remain dormant for long periods, defenders need to inspect multiple components and historical states, leading to substantial computational overhead. As a result, current agent defenses face a severe trade-off among security, utility, and latency.

Direction. Building adaptive cross-component inspection mechanisms is crucial for mitigating computational overhead. A promising direction is to develop backdoor-aware policies that can flexibly inspect selected components and historical states, thereby avoiding unnecessary computational costs and ensuring the practical usability.

6 Conclusion

In this paper, we systematically review various backdoor attacks targeting modern agents. We propose a novel taxonomy that, for the first time, classifies agent backdoors along five dimensions: attack surface, triggering mechanism, attacker capability, attack objective, and persistence and propagation. For defense algorithms, we categorize existing methods according to plan consistency checking, memory sanitization, tool routing, and action monitoring. Finally, we highlight the key challenges in agent backdoor attacks and defenses, aiming to contribute to the development of secure and trustworthy LLM-agent systems.

Limitations

Although this survey systematically reviews backdoor attacks and defense mechanisms targeting intelligent agents, it still has several limitations that need to be considered: (**i**) Current studies mainly focus on single-agent settings, and this survey lacks an in-depth discussion of multi-agent threats. (**ii**) This survey focuses on backdoor attacks and does not provide a detailed discussion of jailbreak attacks or prompt injection attacks.

Ethics Statement

This study investigates backdoor attacks in agent systems and discusses detailed challenges and research directions. Although our goal is to explore the potential security risks of agent systems, we acknowledge that such insights might also be exploited by adversaries. We emphasize that this survey is intended solely to examine these risks and contribute to the development of secure and robust agents.

References

Léo Boisvert, Abhay Puri, Chandra Kiran Reddy Evuru, Nazanin Sepahvand, Nicolas Chapados, Quentin Cappart, Alexandre Lacoste, Krishnamurthy Dj Dvijotham, and Alexandre Drouin. 2025. Malice in agentland: Down the rabbit hole of backdoors in the ai supply chain. *arXiv preprint arXiv:2510.05159*.

Li Changjiang, Liang Jiacheng, Cao Bochuan, Chen Jinghui, and Wang Ting. 2025. Your agent can defend itself against backdoor attacks. *arXiv preprint arXiv:2506.08336*.

Yulin Chen, Haoran Li, Yuan Sui, Yufei He, Yue Liu, Yangqiu Song, and Bryan Hooi. 2025. Can indirect prompt injection attacks be detected and removed? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18189–18206.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. 2024. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *Advances in Neural Information Processing Systems*, 37:130185–130213.

Pengzhou Cheng, Haowen Hu, Zheng Wu, Zongru Wu, Tianjie Ju, Zhuosheng Zhang, and Gongshen Liu. 2025. Hidden ghost hand: Unveiling backdoor vulnerabilities in mllm-powered mobile gui agents. *arXiv preprint arXiv:2505.14418*.

Zhengchunmin Dai, Jiaxiong Tang, Liantao Wu, Peng Sun, and Honglong Chen. 2026. Stateful agent backdoor. *arXiv preprint arXiv:2605.06158*.

Debeshee Das, Julien Piet, Darya Kaviani, Luca Beurer-Kellner, Florian Tramèr, and David Wagner. 2026. Trojan hippo: Weaponizing agent memory for data exfiltration. *arXiv preprint arXiv:2605.01970*.

Edoardo Debenedetti, Ilia Shumailov, Tianqi Fan, Jamie Hayes, Nicholas Carlini, Daniel Fabian, Christoph Kern, Chongyang Shi, Andreas Terzis, and Florian Tramèr. 2025. Defeating prompt injections by design. *arXiv preprint arXiv:2503.18813*.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. 2024. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in Neural Information Processing Systems*, 37:82895–82920.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen Xiang. 2026. Memory injection attacks on llm agents via query-only interaction. *Advances in Neural Information Processing Systems*, 38:46697–46731.

Jing Fang, Saihao Yan, Xueyu Yin, Yinbo Yu, Chunwei Tian, and Jiajia Liu. 2025. Blast: A stealthy backdoor leverage attack against cooperative multi-agent deep reinforcement learning based systems. *arXiv preprint arXiv:2501.01593*.

Yunhao Feng, Yifan Ding, Yingshui Tan, Boren Zheng, Yanming Guo, Xiaolong Li, Kun Zhai, Yishan Li, and Wenke Huang. 2026a. Skilltrojan: Backdoor attacks on skill-based agent systems. *arXiv preprint arXiv:2604.06811*.

Yunhao Feng, Yige Li, Yutao Wu, Yingshui Tan, Yanming Guo, Yifan Ding, Kun Zhai, Xingjun Ma, and Yu-Gang Jiang. 2026b. Backdooragent: A unified framework for backdoor attacks on llm-based agents. *arXiv preprint arXiv:2601.04566*.

Mohamed Amine Ferrag, Abderrahmane Lakas, Norbert Tihanyi, and Merouane Debbah. 2026. Securing llm agents: From prompt sanitization to autonomous red teaming and beyond. *Internet of Things and Cyber-Physical Systems*.

Tongcheng Geng, Zhiyuan Xu, Yubin Qu, and W Eric Wong. 2026. Prompt injection attacks on large language models: A survey of attack methods, root causes, and defense strategies. *Computers, Materials, & Continua*, 87(1).

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Man Hu, Yahui Ding, Yatao Yang, Liangyu Chen, Yanhao Jia, and Shuai Zhao. 2026. Dup: Detection-guided unlearning for backdoor purification in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31059–31067.

Man Hu, Xinyi Wu, Zuofeng Suo, Jinbo Feng, Linghui Meng, Yanhao Jia, Anh Tuan Luu, and Shuai Zhao. 2025. Rethinking reasoning: A survey on reasoning-based backdoors in llms. *arXiv preprint arXiv:2510.07697*.

Dennis Jacob, Emad Alghamdi, Zhanhao Hu, Basel Alomair, and David Wagner. 2025. Better privilege separation for agents by restricting data types. *arXiv preprint arXiv:2509.25926*.

Feiran Jia, Tong Wu, Xin Qin, and Anna Squicciarini. 2025. The task shield: Enforcing task alignment to defend against indirect prompt injection in llm agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29680–29697.

Ruochen Jiao, Shaoyuan Xie, Justin Yue, Takami Sato, Lixu Wang, Yixuan Wang, Qi Alfred Chen, and Qi Zhu. 2025. Can we trust embodied agents? exploring backdoor attacks against embodied llm-based decision-making systems. In *International Conference on Learning Representations*, volume 2025, pages 86638–86668.

Ling-Xin Jin, Wei Jiang, Xiang-Yu Wen, Mei-Yu Lin, Jin-Yu Zhan, Xing-Zhi Zhou, Maregu Assefa Habtie, and Naoufel Werghi. 2025. A survey of backdoor attacks and defences: From deep neural networks to large language models. *Journal of Electronic Science and Technology*, page 100326.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Juhee Kim, Woohyuk Choi, and Byoungyoung Lee. 2025. Prompt flow integrity to prevent privilege escalation in llm agents. *arXiv preprint arXiv:2503.15547*.

Junxian Li, Tu Lan, Haozhen Tan, Yan Meng, and Haojin Zhu. 2026. Slowba: An efficiency backdoor attack towards vlm-based gui agents. *arXiv preprint arXiv:2603.08316*.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on ui control agents, 2024. *URL https://arxiv.org/abs/2406.03679*.

Yige Li, Zhe Li, Wei Zhao, Nay Myat Min, Hanxun Huang, Xingjun Ma, and Jun Sun. 2025. Autobackdoor: Automating backdoor attacks via llm agents. *arXiv preprint arXiv:2511.16709*.

Aishan Liu, Yuguang Zhou, Xianglong Liu, Tianyuan Zhang, Siyuan Liang, Jiakai Wang, Yanjun Pu, Tianlin Li, Junqi Zhang, Wenbo Zhou, and 1 others. 2024a. Compromising embodied agents with contextual backdoor attacks. *arXiv preprint arXiv:2408.02882*.

Shuai Liu, Yiheng Pan, Kun Hong, Ruite Fei, Chenhao Lin, Qian Li, and Chao Shen. 2025. Backdoor threats in large language models—a survey. *Science China Information Sciences*, 68(9):191101.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2024b. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pages 52989–53046.

Yutao Luo, Haotian Zhu, Shuchao Pang, Zhigang Lu, Tian Dong, Yongbin Zhou, and Minhui Xue. 2026. Agentrae: Remote action execution through notification-based visual backdoors against screenshots-based mobile gui agents. *arXiv preprint arXiv:2603.23007*.

Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei-ge Chen, Olga Vrousgos, Corby Rosset, and 1 others. 2024. Agentinstruct: Toward generative teaching with agentic flows. *arXiv preprint arXiv:2407.03502*.

Kanghua Mo, Li Hu, Yucheng Long, and Zhihao Li. 2026. Attractive metadata attack: Inducing llm agents to invoke malicious tools. *Advances in Neural Information Processing Systems*, 38:162770–162793.

Kiourti Panagiota, Wardega Kacper, Susmit Jha, and Li Wenchao. 2020. Trojdrl: Trojan attacks on deep reinforcement learning agents. in proc. 57th acm/ieee design automation conference (dac), 2020, march 2020. In *Proc. 57th ACM/IEEE Design Automation Conference (DAC), 2020*.

Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. 2018. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):180178.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021. Onion: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 9558–9566.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pages 9695–9717.

Jiyang Qiu, Xinbei Ma, Yunqing Xu, Zhuosheng Zhang, and Hai Zhao. 2025. Chain-of-trigger: An agentic backdoor that paradoxically enhances agentic robustness. *arXiv preprint arXiv:2510.08238*.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5977–6043.

Jiawen Shi, Zenghui Yuan, Guiyao Tie, Pan Zhou, Neil Zhenqiang Gong, and Lichao Sun. 2025a. Prompt injection attack to tool selection in llm agents. *arXiv preprint arXiv:2504.19793*.

Tianneng Shi, Jingxuan He, Zhun Wang, Hongwei Li, Linyu Wu, Wenbo Guo, and Dawn Song. 2025b. Progent: Programmable privilege control for llm agents. *arXiv preprint arXiv:2504.11703*.

Tianneng Shi, Kaijie Zhu, Zhun Wang, Yuqi Jia, Will Cai, Weida Liang, Haonan Wang, Hend Alzahrani, Joshua Lu, Kenji Kawaguchi, and 1 others. 2025c. Promptarmor: Simple yet effective prompt injection defenses. *arXiv preprint arXiv:2507.15219*.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce C Ho, Carl Yang, and May Dongmei Wang. 2024. Ehragent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22315–22339.

Saksham Sahai Srivastava and Haoyu He. 2025. Memorygraft: Persistent compromise of llm agents via poisoned experience retrieval. *arXiv preprint arXiv:2512.16962*.

Hang Su, Jun Luo, Chang Liu, Xiao Yang, Yichi Zhang, Yinpeng Dong, and Jun Zhu. 2026. A survey on autonomy-induced security risks in large model-based agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Balachandra Devarangadi Sunil, Isheeta Sinha, Piyush Maheshwari, Shantanu Todmal, Shreyan Mallik, and Shuchi Mishra. 2026. Memory poisoning attack and defense on memory based llm-agents. *arXiv preprint arXiv:2601.05504*.

Guiyao Tie, Jiawen Shi, Pan Zhou, and Lichao Sun. 2026. Badskill: Backdoor attacks on agent skills via model-in-skill poisoning. *arXiv preprint arXiv:2604.09378*.

Bing Wang, Jing Fang, and Shengsheng Qian. 2026. Badinterpreter: Backdoor attack on llm-based interpretable recommendation. *Multimedia Systems*, 32(4):258.

Peiran Wang, Yang Liu, Yunfei Lu, Yifeng Cai, Hongbo Chen, Qingyou Yang, Jie Zhang, Jue Hong, and Ye Wu. 2025. Agentarmor: Enforcing program analysis on agent runtime trace to defend against prompt injection. *arXiv preprint arXiv:2508.01249*.

Xianlong Wang, Hewen Pan, Hangtao Zhang, Minghui Li, Shengshan Hu, Ziqi Zhou, Lulu Xue, Aishan Liu, Yunpeng Jiang, Leo Yu Zhang, and 1 others. 2024a. Trojanrobot: Physical-world backdoor attacks against vlm-based robotic manipulation. *arXiv preprint arXiv:2411.11683*.

Yifei Wang, Dizhan Xue, Shengjie Zhang, and Shengsheng Qian. 2024b. Badagent: Inserting and activating backdoor attacks in llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9811–9827.

Qianshan Wei, Tengchao Yang, Yaochen Wang, Xinfeng Li, Lijun Li, Zhenfei Yin, Yi Zhan, Thorsten Holz, Zhiqiang Lin, and XiaoFeng Wang. 2025. A-memguard: A proactive defense framework for llm-based agent memory. *arXiv preprint arXiv:2510.02373*.

Yuhao Wu, Franziska Roesner, Tadayoshi Kohno, Ning Zhang, and Umar Iqbal. 2024. Isolategpt: An execution isolation architecture for llm-based agentic systems. *arXiv preprint arXiv:2403.04960*.

Zhenlin Xu, Xiaogang Zhu, Yu Yao, Minhui Xue, and Yiliao Song. 2026. From storage to steering: Memory control flow attacks on llm agents. *arXiv preprint arXiv:2603.15125*.

Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. 2024. A comprehensive study of jailbreak attack versus defense for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7432–7449.

Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. 2024. Watch out for your agents! investigating backdoor threats to llm-based agents. *Advances in Neural Information Processing Systems*, 37:100938–100964.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. 2026. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *Advances in Neural Information Processing Systems*, 38:107309–107336.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. 2024. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*.

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pan, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, and 1 others. 2025. A survey on trustworthy llm agents: Threats and countermeasures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6216–6226.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. 2024. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10471–10506.

Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. 2025a. Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents. In *International Conference on Learning Representations*, volume 2025, pages 35331–35366.

Jiwen Zhang, Jihao Wu, Teng Yihua, Minghui Liao, Nuo Xu, Xiao Xiao, Zhongyu Wei, and Duyu Tang. 2024. Android in the zoo: Chain-of-action-thought for gui agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12016–12031.

Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. 2025b. {JBShield}: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 8215–8234.

Wuyang Zhang and Shichao Pei. 2026. Your llm agent can leak your data: Data exfiltration via backdoored tool use. *arXiv preprint arXiv:2604.05432*.

Shuai Zhao, Meihuizi Jia, Zhongliang Guo, Leilei Gan, Xiaoyu Xu, Xiaobao Wu, Jie Fu, Yichao Feng, Fengjun Pan, and Luu Anh Tuan. 2024a. A survey of recent backdoor attacks and defenses in large language models. *arXiv preprint arXiv:2406.06852*.

Shuai Zhao, Qika Lin, Yanhao Jia, Xinyi Wu, Yuwen Li, and Luu Anh Tuan. 2026. Unifle: Uniform fusion of multiple lora experts for backdoor defense in large language models. *IEEE Transactions on Dependable and Secure Computing*.

Shuai Zhao, Luu Anh Tuan, Jie Fu, Jinming Wen, and Weiqi Luo. 2024b. Exploring clean label backdoor attacks and defense in language models. *IEEE/ACM transactions on audio, speech, and language processing*, 32:3014–3024.

Yihe Zhou, Tao Ni, Wei-Bin Lee, and Qingchuan Zhao. 2025. A survey on backdoor threats in large language models (llms): Attacks, defenses, and evaluations. *arXiv preprint arXiv:2502.05224*.

Kaijie Zhu, Xianjun Yang, Jindong Wang, Wenbo Guo, and William Yang Wang. 2025a. Melon: Provable defense against indirect prompt injection attacks in ai agents. *arXiv preprint arXiv:2502.05174*.

Pengyu Zhu, Lijun Li, Yaxing Lyu, Li Sun, Sen Su, and Jing Shao. 2025b. Collaborative shadows: Distributed backdoor attacks in llm-based multi-agent systems. *arXiv preprint arXiv:2510.11246*.

Pengyu Zhu, Zhenhong Zhou, Yuanhe Zhang, Shilinlu Yan, Kun Wang, and Sen Su. 2025c. Demonagent: Dynamically encrypted multi-backdoor implantation attack on llm-based agent. *arXiv preprint arXiv:2502.12575*.

Wei Zou, Mingwen Dong, Miguel Romero Calvo, Shuaichen Chang, Jiang Guo, Dongkyu Lee, Xing Niu, Xiaofei Ma, Yanjun Qi, and Jiarong Jiang. 2026. Poison once, exploit forever: Environment-injected memory poisoning attacks on web agents. *arXiv preprint arXiv:2604.02623*.

A Survey Methodology

We collected candidate papers from ACL Anthology, arXiv, OpenReview, and major AI, security venues using keyword combinations related to *LLM agents*, *backdoor attacks*, *memory poisoning*, *tool poisoning*, *skill backdoors*, *agent security*, *prompt injection*. We included works that study trigger-conditioned or persistent malicious behaviors in agentic systems involving planning, memory, tool use, perception, or multi-agent coordination. We excluded standalone LLM backdoor, jailbreak, and prompt-injection papers unless they directly informed agent-backdoor implantation, activation, evaluation, or defense. Each included work was coded along the five taxonomy axes introduced in the main paper: attack surface, trigger mechanism, attack objective, attacker capability, and persistence/propagation.

B More Backdoor Attacks

Here, we provide supplementary studies on agent backdoor attacks, using the same headings as the main paper to distinguish different types of algorithms.

B.1 Attack Surface

Planner and thought-level Backdoors. Yang et al. (2024) formalized this planner-level vulnerability by distinguishing payloads that corrupt final outputs from payloads that alter only intermediate reasoning while leaving the final response apparently correct. The same work also separated user-query triggers from environment-observation triggers, showing that backdoor activation can occur after the agent receives external feedback rather than only at the initial prompt. In ASB (Zhang et al., 2025a), the Plan-of-Thought backdoor implants poisoned demonstrations into the system prompt and couples them with trigger-bearing user prompts, making the agent's reasoning trajectory itself part of the attack channel. Stateful Agent Backdoor (Dai et al., 2026) provides an even more explicit temporal formulation by modeling the backdoor as a finite-state machine whose malicious behavior unfolds across sessions after a one-time trigger injection.

Memory and Retrieval Backdoors. MemoryGraft (Srivastava and He, 2025) shifts the target from isolated factual records to successful experience traces, exploiting the agent's tendency to imitate retrieved successful procedures in later semantically similar tasks. Forever (Zou et al., 2026) introduces environment-injected trajectory-based agent memory poisoning, showing that a single contaminated observation can poison agent memory and later trigger malicious behavior across websites and sessions, even without direct memory access. Memory control flow attacks (Xu et al., 2026) show that retrieved memories can dominate later tool-selection and execution control flow, forcing unintended tool usage even against explicit user instructions. Trojan Hippo (Das et al., 2026) reframes persistent memory compromise as a confidentiality threat by planting dormant payloads that activate when the user later discusses sensitive topics such as finance, health, or identity.

Tool Interface, Skill and Protocol-Layer Backdoors. SkillTrojan (Feng et al., 2026a) shifts compromise from metadata to executable skill implementations by embedding malicious logic in plausible reusable skills (Wang et al., 2026). It partitions encrypted payload fragments across benign-looking skill invocations and activates them only under predefined triggers, showing that the malicious payload need not be visible in any single skill call. BadSkill (Tie et al., 2026) presents model-in-skill supply-chain attack, where third-party skills package backdoored models triggered by semantic combinations of otherwise routine skill parameters. Collaborative shadows (Zhu et al., 2025b) extends this logic to multi-agent and multi-tool systems by decomposing a backdoor into distributed primitives that remain dormant in isolation and activate only when agents collaborate in a specific sequence. Back-Reveal (Zhang and Pei, 2026) further shows that tool-use backdoors can target data exfiltration by making a triggered agent retrieve stored user context and leak it through disguised retrieval tool calls.

Executor and Perception Modules. BALD (Jiao et al., 2025) systematically studies backdoors against embodied LLM-based decision-making systems through word injection, scenario manipulation, and knowledge injection, covering autonomous-driving and home-robot tasks. TrojanRobot (Wang et al., 2024a) demonstrates physical-world backdoors against VLM-based robotic manipulation by embedding a backdoor module into the visual perception pathway of modular robotic policies. Contextual backdoor attacks against embodied agents show that poisoned contextual demonstrations can generate programs that look logically sound but contain defects activated by

Table 1: Comparison with related survey lines. ✓: central focus; △: partial or background coverage; ✗: not a main focus.

Survey line	Main emphasis	Backdoor centric	Agentic surfaces	Agent backdoor threat model	Multi-axis taxonomy	Persistence & prop.	Agent-level defenses	Evaluation & future directions
LLM backdoor surveys	Poisoning, triggers, and defenses for standalone LLMs or NLP models; agentic systems are usually outside the main threat model (Zhao et al., 2024a; Liu et al., 2025; Zhou et al., 2025).	✓	✗	✗	△	✗	△	△
Prompt-injection / jailbreak surveys	Instruction-level attacks against LLM applications, including indirect prompt injection; backdoors are usually treated as adjacent threats rather than the central object (Xu et al., 2024; Geng et al., 2026; Yi et al., 2024).	✗	△	✗	△	△	△	△
General LLM-agent security / trust surveys	Broad security, reliability, privacy, and tool-use risks in LLM agents, often covering many threat types beyond backdoors (Yu et al., 2025).	△	✓	△	△	△	△	✓
Layered agent-security surveys	Component- or layer-based analyses of agent systems, such as planning, memory, tools, execution, and multi-agent collaboration (Ferrag et al., 2026; Su et al., 2026).	△	✓	△	△	△	✓	✓
This survey	**Backdoor threats and defenses in intelligent agents, organized by attack surface, trigger mechanism, attack objective, attacker capability, and persistence/propagation.**	✓	✓	✓	✓	✓	✓	✓

later textual or visual triggers (Liu et al., 2024a). In mobile GUI agents, AgentGhost (Cheng et al., 2025) identifies interaction-level triggers such as historical steps, environment states, and task progress, and reports high attack accuracy with small utility degradation in evaluated mobile benchmarks.

B.2 Trigger Mechanisms

Lexical and Semantic Triggers. AgentPoison optimizes retrieval-space triggers so that triggered inputs retrieve malicious memory demonstrations while benign inputs preserve normal behavior (Chen et al., 2024). BadSkill activates on semantic combinations of normal-looking skill parameters rather than on a single rare string (Tie et al., 2026). Trojan Hippo delays activation until user conversations enter sensitive topical regions, which makes the trigger appear natural in ordinary memory-augmented assistants (Das et al., 2026).

Multimodal and Visual Triggers. AgentRAE (Luo et al., 2026) uses benign app icons in notifications as visually natural triggers for remote action execution . SlowBA (Li et al., 2026) uses realistic pop-up windows as GUI triggers to activate latency-oriented backdoors rather than wrong-action payloads . Chain-of-Trigger (Qiu et al., 2025) also validates its ordered trigger mechanism on VLM-based agents, showing that multi-step triggers can generalize beyond purely textual agents.

Stateful and Chained Triggers. Chain-of-Trigger relies on an ordered trigger sequence in which the initial trigger is supplied first and later triggers are drawn from the environment, enabling long-horizon agentic control (Qiu et al., 2025). Poison Once, Exploit Forever demonstrates that one contaminated observation can poison a web agent's memory and activate in future tasks on different websites, making the trigger both cross-session and cross-site (Zou et al., 2026). Memory control flow attacks exploit the repeated read-memory, plan, execute, and write-memory loop to make memory retrieval steer future control flow over long interaction horizons (Xu et al., 2026).

Distributed Triggers. BLAST foreshadows the same principle in cooperative multi-agent deep reinforcement learning by showing that a compromised agent can leverage team interactions to produce backdoor effects (Fang et al., 2025).

B.3 Attack Objectives

Integrity & Confidentiality. Confidentiality could be treated as a central objective because tool and memory access allow a triggered agent to retrieve, transform, and leak sensitive user data. Trojan Hippo (Das et al., 2026) is explicitly designed for deferred personal-data exfiltration from persistent memory systems, with dormant payloads that

activate after many benign sessions when sensitive topics appear. Back-Reveal embeds semantic triggers into fine-tuned tool-use agents and uses memory-access tool calls and disguised retrieval tool calls to exfiltrate stored user context (Zhang and Pei, 2026). Collaborative Shadows includes distributed exfiltration as one of the targeted attacks enabled by multi-agent collaboration sequences (Zhu et al., 2025b). Malice in Agentland further shows that poisoned agent-training traces can cause trigger-conditioned leakage of confidential user information after downstream fine-tuning (Boisvert et al., 2025). These studies show that memory- and tool-augmented agents turn backdoors into privacy and data-governance threats, not only correctness threats.

Auditing Evasion and Stealth. Chain-of-Trigger (Qiu et al., 2025) demonstrates near-perfect attack success, a near-zero false-trigger rate, and improved robustness on benign tasks, making the attack harder to distinguish from ordinary improvements in agent robustness. Attractive metadata attack (Mo et al., 2026) keeps tool metadata syntactically and semantically valid while still biasing invocation, which makes it stealthy under prompt-level defenses and structured tool-selection protocols. AgentGhost (Cheng et al., 2025) and AgentRAE (Luo et al., 2026) optimize stealth through natural interaction-level or iconographic triggers in GUI settings, while BadSkill uses routine skill parameters as semantic triggers in a model-in-skill supply-chain setting (Tie et al., 2026).

C More Defenses

C.1 Input and Context Sanitization

The first line of defense is to prevent triggers from ever reaching the agent's reasoning core. Two threads dominate. The first carries over from classical LLM backdoors: token-level filtering, exemplified by ONION-style perplexity filtering and back-translation, which flag tokens that contribute disproportionately to model perplexity (Qi et al., 2021). These methods are inexpensive and serve as standard baselines in agent backdoor evaluations (Changjiang et al., 2025), but they are reliably defeated by semantically natural triggers (Chen et al., 2024; Qiu et al., 2025) and by trigger-free attacks (Srivastava and He, 2025), and are therefore insufficient on their own (Jin et al., 2025).

The second thread is provenance-aware delimiting. Spotlighting marks every untrusted token in the context with delimiters, datamarks, or encodings so that the model can distinguish trusted user instructions from untrusted retrieved content. PromptArmor extends detection-based defenses for indirect prompt injection by using an off-the-shelf LLM as a guardrail to detect and remove injected instructions from agent inputs or tool outputs before the backend agent processes them (Shi et al., 2025c).

Input-side defenses are necessary but not sufficient for backdoors. By definition, a backdoor's trigger is benign in isolation and therefore evades content-based filters. The literature has accordingly shifted toward defenses that inspect the agent's behavior rather than its inputs.

C.2 Plan Consistency Checking

When triggers do reach the planner, an agent can be made to interrogate its own reasoning. ReAgent is the most direct example: it observes that backdoor attacks often produce inconsistencies among the user's instruction, the agent's planning, and its execution, and exploits this observation with a two-level verifier (Changjiang et al., 2025). At the execution level, ReAgent checks whether each action follows logically from the immediately preceding thought; at the planning level, it asks the agent's own LLM to reconstruct the original user instruction from its thought trajectory and compares the reconstruction to the actual instruction. This forces the attacker into a dilemma: embedding the backdoor only in actions exposes the execution-level mismatch, while embedding it in both thoughts and actions exposes the planning-level mismatch.

Task Shield enforces a related but stricter property: every instruction the agent follows and every tool it invokes must demonstrably serve the user's stated objective (Jia et al., 2025). The closely related plan-consistency line of work tasks an auxiliary LLM with auditing the alignment between plan and goal at every step; these methods share with ReAgent the assumption that the attacker cannot simultaneously satisfy multiple consistency constraints. Plan-level verification has not yet been evaluated against CoTri-style attacks that explicitly preserve clean-task utility (Qiu et al., 2025).

C.3 Memory Sanitization

Memory creates the longest-lived attack surface in an agent, as a single poisoned record can influence behavior across sessions or even over months. Defenses should therefore operate both when memory

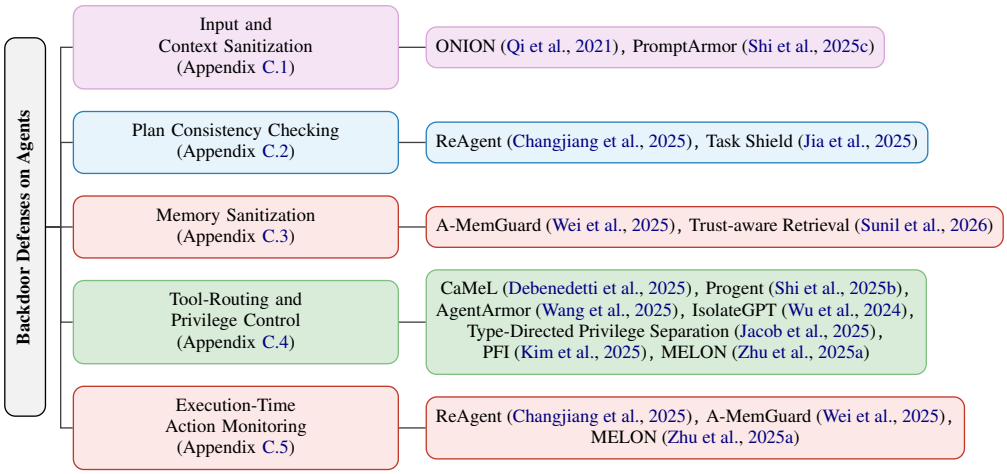

Figure 3: Overview of agent backdoor defenses categorized by input sanitization, plan consistency checking, memory sanitization, tool routing and privilege control, and action monitoring.

is written and when it is retrieved. A-MemGuard (Wei et al., 2025) is the first defense framework designed specifically for LLM-agent memory backdoors. It identifies two properties that make memory backdoors hard to address: context-conditional activation, which evades input filters, and self-reinforcing error cycles, in which a poisoned retrieval shapes future writes that further entrench the compromise. A-MemGuard addresses both with two mechanisms: consensus retrieval, which requires multiple semantically diverse retrievals to agree before any one record is used, and a dual-memory feedback loop that distills flagged entries into negative "lessons" consulted before future actions.

Trust-aware retrieval (Sunil et al., 2026) approaches the same problem with a two-stage gating pipeline. At the append stage, candidate memory entries pass through static heuristics, an LLM classifier, and an optional sandbox; at the retrieve stage, a composite trust score with temporal decay weights every entry before top-k selection. Their evaluation in an electronic-health-records agent setting establishes an important empirical caveat: trust thresholds tuned too tightly block legitimate entries, while thresholds tuned too loosely miss subtle attacks, and the optimal operating point depends heavily on whether the memory store already contains pre-existing legitimate entries against which a MINJA-style attack must compete. This finding tempers MINJA's headline numbers and clarifies that realistic deployment conditions reduce attack effectiveness even before any specific defense is applied.

Neither A-MemGuard nor trust-aware retrieval has been evaluated against MemoryGraft's trigger-free poisoning (Srivastava and He, 2025) or against Memory Control Flow Attacks (Xu et al., 2026), these remain open empirical questions. A complementary direction, articulated in the MemoryGraft discussion, calls for cryptographic provenance attestation on memory writes so that the source of every record can be traced.

C.4 Tool-Routing and Privilege Control

Tool-routing defenses operate not by inspecting content but by restricting capability. The premise is that no amount of input sanitization can be sound when an attacker controls some retrieved content, so the safest architecture is one in which retrieved content cannot, by construction, trigger high-privilege actions.

CaMeL (Debenedetti et al., 2025) realizes this principle by extracting the control flow from the trusted user query into a typed program. Untrusted data then flows in as opaque variables that cannot redirect control. Progent (Shi et al., 2025b) offers a complementary mechanism: a domain-specific language for programmable privilege control, with policies (either hand-written or LLM-generated) enforced deterministically at every tool call. AgentArmor (Wang et al., 2025) approaches the problem from a program-analysis perspective. It converts the agent's runtime trace into a Program Dependence Graph, annotates nodes with security types, and enforces information-flow policies before each tool call, substantially reducing the average attack success rate on AgentDojo.

Architectural isolation represents an earlier line of defense related to these typed-flow approaches.

IsolateGPT (Wu et al., 2024) implements a hub-and-spoke architecture in which each tool or third-party application runs in its own sandboxed spoke with its own LLM instance, and a central hub mediates inter-spoke flow. It preserves functionality on three-quarters of tested queries with under 30% performance overhead. Type-directed privilege separation (Jacob et al., 2025) generalizes the dual-LLM pattern by allowing the privileged LLM to read structured data types (integers, enumerations) from the quarantined LLM but never freeform text. PFI (Kim et al., 2025) combines untrusted-data identification, least-privilege enforcement, and data-flow validation, reporting higher Secure Utility Rate than IsolateGPT and dual-LLM baselines. For the deployment-time activation phase of a backdoor in particular, MELON (Zhu et al., 2025a) provides a runtime test: re-execute the agent trajectory with a masked user prompt. If the agent still produces the same tool calls, those calls must originate from injected content rather than from the user's task.

C.5 Execution-Time Action Monitoring

Execution-time action monitoring is the final stage for defending against backdoor attacks. At this stage, the internal state induced by the trigger condition is transformed into tool calls or decisions. Therefore, once a backdoor is activated, the defense should not merely filter suspicious tokens or retrieved contexts, but should also examine whether the actions proposed by the agent are consistent with the user goal, the execution plan, and the permitted privilege boundaries (Ferrag et al., 2026).

For example, ReAgent (Changjiang et al., 2025) is designed to protect LLM-based agents against backdoor attacks. It observes that backdoor activation often introduces inconsistencies among the user instruction, the agent's reasoning trajectory, and the executed actions. To address this issue, ReAgent introduces a two-level verifier. At the execution level, it checks whether each action logically follows the preceding thoughts. At the planning level, it reconstructs the original user instruction from the agent's reasoning trajectory and compares it with the actual instruction. This design forces the attacker into a dilemma: attacks targeting actions lead to execution inconsistency, while attacks targeting reasoning expose planning inconsistency.

In addition, A-MemGuard (Wei et al., 2025) shows that poisoned memory can induce malicious behavior only after it is retrieved and incorporated into subsequent decision making. Therefore, action monitoring should not be separated from memory trust estimation. For tool invocation, MELON (Zhu et al., 2025a) masks the user prompt and re-executes the agent trajectory. If the same tool call is still produced, the action is likely driven by injection or untrusted content rather than by the user's task. However, execution-time monitoring for agent backdoors remains underexplored. Existing monitors often assume that malicious behavior causes obvious inconsistencies, but stealthy backdoors may preserve the quality of the final answer, imitate benign reasoning, or distribute the attack across multiple tools, memories, or agents. Therefore, future defenses need to assess not only whether an action is safe in isolation, but also whether it remains trustworthy across long-term, cross-session, and multi-agent trajectories. The taxonomy structure is shown in Figure 3.

Summary and Challenges

Defenses against agent backdoors have evolved from input-side sanitization into a multi-layer defense system that covers the full lifecycle of agents. Although component-level defenses have made progress, agent defense still faces a severe security-utility-latency trade-off. Most defense algorithms introduce substantial API overhead and inference latency. A lightweight defense architecture with global tracing capability remains lacking.

D Benchmarks and Evaluation

Evaluation for agent backdoors remains fragmented. Unlike conventional backdoor attacks, where the poisoned model is usually tested on clean and triggered inputs, agent backdoors must be evaluated over multi-step workflows involving planning, memory retrieval, tool selection, environment observation, and execution. Therefore, it is important to distinguish reusable benchmarks from attack methods. A benchmark provides shared tasks, environments, and scoring protocols, while an attack method may only introduce a method-specific evaluation setup. Table 2 summarizes representative reusable benchmarks and method-specific evaluation resources for agent-security and agent-backdoor research.

Existing agent-backdoor evaluations do not rely on a single standard corpus. Instead, datasets are usually selected according to the attacked agent surface. Tool-use attacks often use tool-selection or tool-trajectory resources such as ToolBench (Qin et al., 2024), while task-level agent backdoors use AgentInstruct (Mitra et al., 2024) and its task subsets, including Operating System (Liu et al., 2024b), Mind2Web (Deng et al.,

2023), and WebShop (Yao et al., 2022). Memory and RAG-based attacks commonly rely on long-term retrieval or knowledge-intensive settings such as StrategyQA (Geva et al., 2021), and EHRAgent-style (Shi et al., 2024) healthcare records, including EHR datasets such as MIMIC-III (Johnson et al., 2016) and eICU (Pollard et al., 2018). GUI and embodied-agent attacks further require visual or interaction-based datasets such as AndroidControl (Li et al.) and AITZ (Zhang et al., 2024), as well as domain-specific autonomous-driving and home-robot tasks. This surface-dependent dataset usage explains why results are difficult to compare across papers: reusable benchmarks such as ASB (Zhang et al., 2025a), AgentDojo (Debenedetti et al., 2024), and InjecAgent (Zhan et al., 2024) provide standardized protocols, but are not always backdoor-native, whereas backdoor-specific resources better capture trigger activation and persistence but are often tied to one attack surface or one paper. A mature benchmark should therefore jointly measure attack success rate (ASR), clean utility, false-trigger behavior, cross-session persistence, distributed propagation, and runtime or defense overhead.

The most reusable benchmark is Agent Security Bench (ASB) (Zhang et al., 2025a), which formalizes attacks and defenses for LLM-based agents across 10 scenarios, 10 agents, more than 400 tools, 23 attack/defense methods, and 8 metrics. ASB is not limited to backdoors, but it explicitly includes memory poisoning, a Plan-of-Thought backdoor attack, and mixed attacks. AgentDojo (Debenedetti et al., 2024) and InjecAgent (Zhan et al., 2024) are adjacent security benchmarks rather than backdoor-native benchmarks. AgentDojo evaluates prompt injection attacks and defenses in tool-using agents with 97 realistic tasks and 629 security test cases, using metrics such as utility, utility under attack, and targeted ASR. InjecAgent evaluates indirect prompt injection in tool-integrated agents with 1,054 test cases, 17 user tools, and 62 attacker tools, and separates direct harm from private-data exfiltration. Although these benchmarks do not model persistent backdoor implantation directly, they provide useful evaluation protocols for tool hijacking, exfiltration, and security–utility trade-offs.

Backdoor-oriented evaluation resources and method-specific benchmarks have also begun to appear, but most are still introduced within individual attack papers rather than as unified community benchmarks. BackdoorAgent (Feng et al., 2026b) constructs a stage-aware benchmark over Agent QA, Agent Code, Agent Web, and Agent Drive, and evaluates planning-stage, memory-stage, and tool-stage backdoors through ASR and trigger persistence. DemonAgent (Zhu et al., 2025c) introduces AgentBackdoorEval for evaluating stealthy and audit-evasive agent backdoors, while SkillTrojan (Feng et al., 2026a) releases more than 3,000 curated backdoored skills for skill-level supply-chain evaluation. Collaborative Shadows (Zhu et al., 2025b) further introduces a multi-agent benchmark for distributed backdoors, where malicious primitives activate only through specific collaboration sequences. For domain-specific settings, BALD (Jiao et al., 2025) evaluates embodied-agent backdoors in autonomous-driving and home-robot tasks, while Trojan Hippo (Das et al., 2026) provides a dynamic framework for persistent memory attacks and data exfiltration across heterogeneous memory backends.

Summary and Limitations

ASR remains the core metric, but it is insufficient on its own. It cannot fully capture clean utility, accidental activation, persistence, or defense cost. Existing evaluations often rely on method-specific settings rather than unified community benchmarks, making results difficult to compare across attack surfaces. A standardized benchmark that jointly measures ASR, benign utility, false-trigger behavior, cross-session persistence, distributed propagation, and runtime overhead is still lacking.

Table 2: Representative benchmarks and evaluation resources for agent-security and agent-backdoor research. Reusable benchmarks such as ASB, AgentDojo, and InjecAgent provide shared evaluation protocols, whereas most backdoor-specific evaluations remain surface-specific and are introduced within individual attack papers.

Resource	**Type**	**Agent surface**	**Evaluation focus**	**Main metrics**
ASB (Zhang et al., 2025a)	Reusable agent-security benchmark	Prompts, tools, memory, planning	General agent attacks and defenses, including memory poisoning, Plan-of-Thought backdoors, and mixed attacks	ASR, utility, RR, FPR/FNR, defense effectiveness
AgentDojo / InjecAgent (Debenedetti et al., 2024; Zhan et al., 2024)	Reusable tool-agent security benchmarks	Tool use, external content, user/attacker tools	Indirect prompt injection, malicious tool execution, and private-data exfiltration	Clean utility, utility under attack, targeted ASR, exfiltration success
BackdoorAgent (Feng et al., 2026b)	Backdoor-native framework	Planning, memory, tool-use stages	Stage-specific backdoors implanted into agent workflows	ASR, clean utility, trigger persistence, stage-wise effectiveness
Memory-backdoor protocols (Chen et al., 2024; Dong et al., 2026; Srivastava and He, 2025; Zou et al., 2026; Das et al., 2026)	Surface-specific protocols	Long-term memory, RAG, experience memory, web-agent memory	Poisoned retrieval, query-only memory injection, trigger-free experience poisoning, and dormant memory payloads	Retrieval success, ASR, poison rate, cross-session persistence, exfiltration success
Skill-backdoor protocols (Feng et al., 2026a; Tie et al., 2026)	Supply-chain suites	Third-party skills, model-in-skill deployments	Skill-level triggers, semantic parameter triggers, and malicious skill composition	ASR, clean utility, poison rate, trigger specificity
Multi-agent backdoor protocols (Zhu et al., 2025b)	Distributed-backdoor framework	Inter-agent communication and collaboration	Fragmented malicious primitives activated only through specific collaboration sequences	ASR, benign utility, distributed activation, propagation success
Embodied / GUI-agent protocols (Jiao et al., 2025; Cheng et al., 2025; Luo et al., 2026; Li et al., 2026)	Domain-specific protocols	Embodied perception, screenshots, notifications, GUI execution	Visual, environmental, interaction-history, notification-based, and latency-oriented triggers	ASR, action success, clean utility, stealth, latency overhead