A Survey on Agentic Security: Applications, Threats and Defenses

Anonymous ACL submission

Abstract

We present the first holistic survey of the agentic security landscape, structuring the field around three interdependent pillars: Applications, Threats, and Defenses. We provide a comprehensive taxonomy of over 150 papers, explaining how agents are used, the vulnerabilities they possess, and the countermeasures designed to protect them. A detailed cross-cutting analysis shows emerging trends in agent cardinality and architecture while revealing critical research gaps in coverage.

1 Introduction

Since their introduction, Large Language Models (LLMs) have been used in the domain of cybersecurity (Xu et al., 2025a; Wang et al., 2024; Deng et al., 2024a). The transition of research landscape from passive LLMs to autonomous LLM-agents (Yao et al., 2023; Shinn et al., 2023; Schick et al., 2023) has made these models significantly more capable, allowing them not just to describe a solution but to execute it.

Definition: LLM Agent. *For this paper, we define an* **LLM Agent** *as a computational system whose core decision module is an LLM that plans, invokes tools/APIs, and acts in an external environment while observing feedback and adapting subsequent actions. It maintains state (short/long-term memory or a knowledge store) and may include explicit self-critique/verification and governance layers to satisfy task goals and safety constraints.*

This newfound agency has enabled LLM-agents to demonstrate remarkable capabilities across the security spectrum (Shen et al., 2025; Zhu et al., 2025b; Lin et al., 2025b). However, a number of studies have shown that the very act of wrapping an LLM in an agentic framework significantly increases its vulnerability (Saha et al., 2025; Kumar et al., 2025; Chiang et al., 2025). In response, a growing body of research has focused on developing countermeasures to harden these systems (Debenedetti et al., 2025; Udeshi et al., 2025).

The rapid development of agentic security research – with over 150 papers in 2024-2025 alone – has created a fragmented landscape that lacks comprehensive analysis. While existing surveys provide valuable insights into specific aspects like security threats (Deng et al., 2024c), trustworthiness (Yu et al., 2025), enterprise governance (Raza et al., 2025) and core LLM safety (Ma et al., 2025), they fail to capture the complete picture, as shown in Table 1. This fragmentation leaves practitioners and researchers without a unified framework for understanding how agent capabilities, vulnerabilities, and defenses interconnect.

In this work we present the first holistic survey of the agentic security landscape, structured to answer three key questions a security researcher would ask: *"What can agents do for my security?"* (Applications), *"How can they be attacked?"* (Threats), and *"How do I stop them?"* (Defenses). To this end, we define three pillars of taxonomy:

Applications (§2). Using LLM-agents in downstream cybersecurity tasks, including red teaming (autonomous vulnerability discovery), blue teaming (defending against threats), and domain-specific security (cloud, web).

Threats (§3). Security vulnerabilities inherent to agentic systems that attackers can exploit.

Defenses (§4). Techniques and countermeasures used to harden agentic systems against the threats.

By uniquely bridging these three pillars, our survey provides a complete picture of the current state of the art, transforming a scattered collection of individual research efforts into an actionable body of knowledge. Our contributions are threefold:

- **Holistic review.** We conduct an in-depth survey of agentic security through a comprehensive three-pillar taxonomy, as presented in Fig. 1.
- **Focus on applications.** We provide a detailed

Table 1: Survey comparison. Legend: ✓ = covered; △ = partial/limited; ✗ = not covered. A: Applications, T: Threats, D: Defenses, B: Benchmarks/testbed surveyed.

Survey	A	T	D	B	Focus
Yu et al. (2025)	✗	✓	✓	✓	Trustworthiness
Raza et al. (2025)	✗	△	△	✗	Enterprise governance & risk
Deng et al. (2024c)	✗	✓	△	✗	Security threats
He et al. (2024)	✗	✓	✓	✗	Technical vulnerabilities
Ma et al. (2025)	△	✓	✓	△	Large-model safety (agents as subset)
This Survey	✓	✓	✓	✓	**Holistic agentic security**

review focused on how agents are actually used by security teams—covering offensive, defensive, and domain-specific tasks, an area largely overlooked by prior surveys.

- **Cross-cutting analysis.** We analyze 150+ papers to identify key trends and critical gaps— for example, the migration from monolithic to planner-executor and multi-agent architectures, almost exclusive focus on single commercial LLM (GPT), uneven threat and modality coverage (RAG poisoning under-defended, few works on images), and benchmark fragmentation.

2 Applications of Agents in Security

2.1 Offensive Security Agents (Red-Teaming)

2.1.1 Autonomous Penetration Testing

This area explores agents that autonomously perform end-to-end penetration testing such as reconnaissance, exploitation, and lateral movement—using adaptive planning and feedback. Deng et al. (2024a) propose PentestGPT, the first LLM-based framework with a Reasoning–Generation–Parsing design reducing context loss, while Shen et al. (2025) and Kong et al. (2025b) extend this with multi-agent RAG and task-graph coordination respectively. Nieponice et al. (2025) introduce an SSH-focused system, and Happe and Cito (2025a) show autonomous adaptation of LLM agents in enterprise testbeds. Evaluation works include HackSynth (Muzsai et al., 2024) and AutoPentest (Henke, 2025), which benchmark or integrate continuous exploit intelligence. Comparative and system-focused studies include Happe and Cito (2025b) on interactive vs autonomous agents, Singer et al. (2025) introducing Incalmo for reliable multi-host execution, and Luong et al. (2025), who achieves state-of-the-art results on AutoPenBench (Gioacchini et al., 2024) and AI-Pentest-Benchmark (Isozaki et al., 2024).

2.1.2 Automated Vulnerability Discovery & Fuzzing

This area studies agents that use reasoning to guide fuzzing like identifying bug-prone regions, generating targeted inputs, and adapting from feedback. Zhu et al. (2025a) propose Locus for deep-state exploration via predicate synthesis. Meng et al. (2024) extract protocol grammars to guide fuzzing, while Fang et al. (2024) show LLMs can autonomously exploit one-day flaws. Zhu et al. (2025d) extend this to multi-agent zero-day discovery.Wang and Zhou (2025) present a two-phase agentic system for Android vulnerability discovery and validation. Lee et al. (2025b); Zhu et al. (2025b); Wang et al. (2025c) introduce benchmarks to evaluate LLM agents on tasks like exploitation and repair, while ExCyTInBench(Wu et al., 2025b) highlights challenges in multi-step reasoning. LLMFuzzer (Yu et al., 2024), TitanFuzz(Deng et al., 2023), and FuzzGPT (Deng et al., 2024b) use fuzzing or reasoning-based input generation.

2.1.3 Exploit (Malware) Generation & Adaptation

This area examines agents that generate environment-aware exploits or malware, often polymorphic and evasive. Lupinacci et al. (2025) show that LLM agents can be coerced into autonomously executing malware via prompt injection, RAG backdoors, and inter-agent trust abuse. Saha and Shukla (2025) proposed a multi-agent system producing diverse malware samples for studying evasion tactics. He et al. (2025a) describe an Agent-in-the-Middle attack injecting malicious logic into multi-agent frameworks by intercepting and mutating messages, while Ullah et al. (2025) introduce CVE-Genie, which rebuilds CVE environments and reproduces 51% of 841 exploits. Finally, Fakih et al. (2025) present a repair system combining fine-tuned LLMs and iterative validation to generate accurate vulnerability patches.

2.2 Defensive Security Agents (Blue-Teaming)

2.2.1 Autonomous Threat Detection & Incident Response

This area studies agentic SOC frameworks that monitor alerts, analyze threats, and execute re-

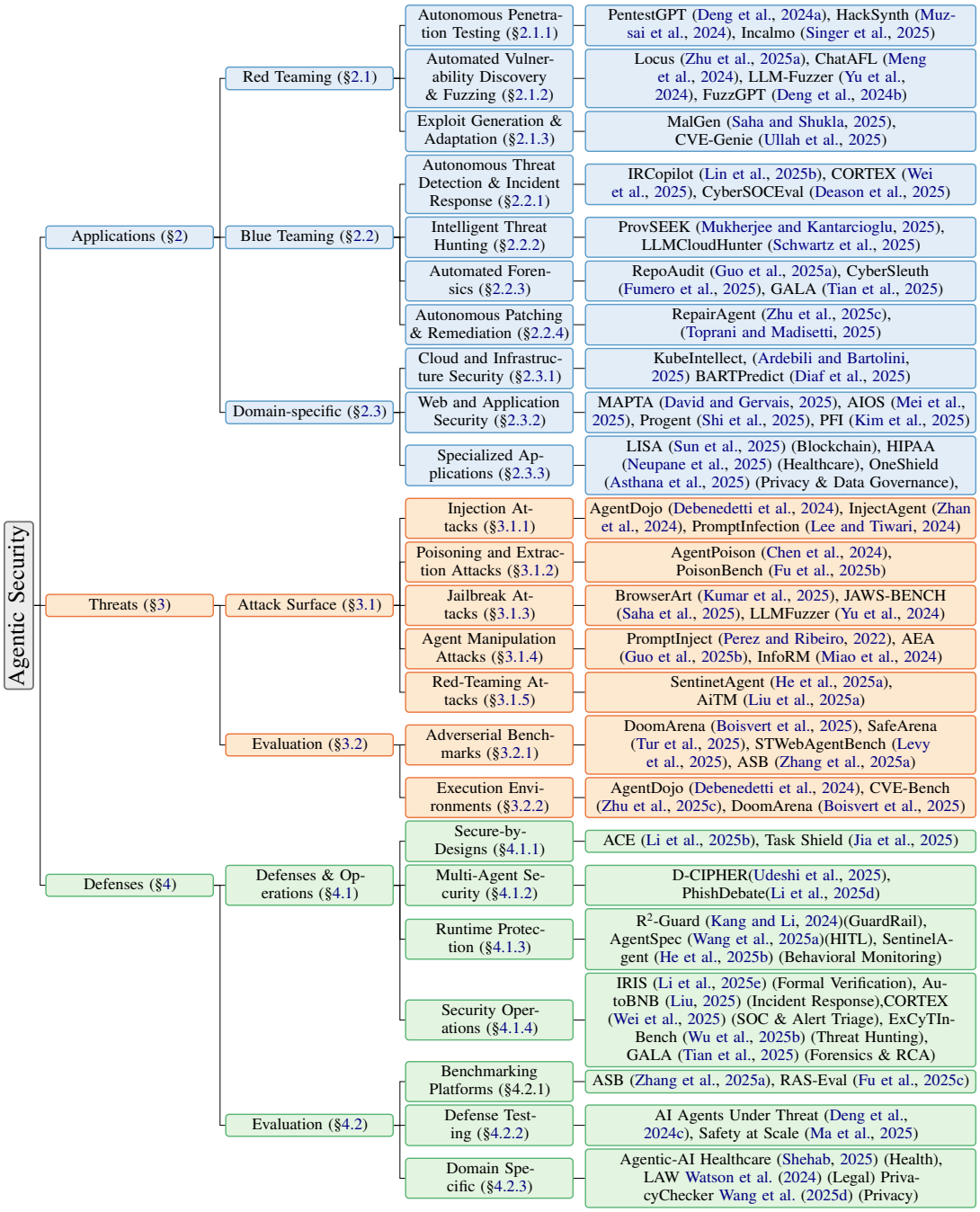


Figure 1: Overview of Agentic Security Taxonomy

sponse playbooks. Tellache et al. (2025) propose a RAG-based agent combining CTI and SIEM data for automated triage, while Lin et al. (2025b) and Wei et al. (2025) propose IRCopilot, with role-based agents for incident response, and CORTEX, a collaborative agent system reducing false positives. Liu (2025) explore centralized and hybrid agent models for team-based response, while Singh et al. (2025) find LLMs are mainly used as assistive tools in real-world SOCs. (Deason et al., 2025) benchmarks LLM threat reasoning, exposing performance gaps, while Molleti et al. (2024) survey log-analysis agents and highlight scalability and robustness challenges.

2.2.2 Intelligent Threat Hunting

This category studies proactive agents that analyze unstructured threat intelligence, form hypotheses, and query system data to uncover hidden adversaries like APTs. Mukherjee and Kantarcioglu (2025) present a provenance-forensics framework using RAG, chain-of-thought reasoning, and agent orchestration to refine and verify evidence, im-

proving precision and recall over SOTA baselines. Meng et al. (2025a) analyze failures in LLM-assisted CTI workflows and propose fixes for contradictions and generalization gaps. Schwartz et al. (2025) introduce LLMCloudHunter, extracting cloud IoCs and generating high-precision detection rules convertible for platforms like Splunk.

2.2.3 Automated Forensics & Root Cause Analysis

This category examines agents that reconstruct attacks, trace entry points, and generate verifiable forensic reports.Guo et al. (2025a) introduce a repository-level auditing agent that uses memory and path-condition checks to reduce hallucinations. Alharthi and Yasaei (2025) and Fumero et al. (2025) develop LLM-powered tools for classifying logs, extracting forensic intelligence, and analyzing network traffic, while Tian et al. (2025) combine causal inference and LLM reasoning for iterative root-cause analysis.Pan et al. (2025) present the MAST taxonomy of multi-agent failure modes and an LLM-as-Judge pipeline for execution failure detection, while Alharthi and Garcia (2025) introduce CIAF, an ontology-driven framework for structuring cloud logs and assembling incident narratives.

2.2.4 Autonomous Patching & Vulnerability Remediation

This category explores agents that detect and remediate vulnerabilities through automated patch synthesis and validation. Zhu et al. (2025c) benchmark LLM agents on real CVE repair using static-analysis tools, while Bouzenia et al. (2025) introduce RepairAgent, an autonomous pipeline that achieves state-of-the-art results on Defects4J. Applied systems include Google's Gemini-based patching workflow (Keller and Nowakowski, 2024) and an IaC agent (Toprani and Madisetti, 2025) that integrates automated remediation into CI/CD pipelines.

2.3 Domain-specific Applications

2.3.1 Cloud and Infrastructure Security

This section covers agents securing cloud and infrastructure through automated scanning, hardening, and remediation. Yang et al. (2025b) propose a two-phase workflow—sandbox "exploration" followed by verified "exploitation"—to safely test CSPM remediations. Ardebili and Bartolini (2025) introduce an LLM supervisor coordinating subagents for log analysis, RBAC auditing, and debugging, while Ye et al. (2025) present LLMSecConfig, combining static analysis and RAG to fix Kubernetes misconfigurations. Diaf et al. (2025) propose BARTPredict for IoT traffic forecasting and anomaly detection, and Toprani and Madisetti (2025) describe an IaC-focused agent that autogenerates CI/CD-ready fixes for policy-compliant hardening.

2.3.2 Web and Application Security

This area investigates agentic systems for secure automation in web and OS environments. David and Gervais (2025) propose a multi-agent web pentesting framework with sandboxed PoC validation for safe, repeatable exploit testing. Mudryi et al. (2025) analyze browser-agent threats like prompt injection and credential leaks, introducing layered defenses like sanitization and formal analysis. At the OS level, Mei et al. (2025) design an agent-oriented OS that isolates LLMs and mediates tool access via policy. Hu et al. (2025) formalize OS agent observation/action spaces to support structured risk analysis. Kim et al. (2025) validate control/data flows to prevent privilege escalation, while Shi et al. (2025) present Progent, a runtime enforcing deterministic, fine-grained permissions that eliminate attack success in red-team evaluations.

2.3.3 Specialized Applications

This section reviews agentic security across finance, healthcare, privacy, and embodied systems. In finance, Sun et al. (2025) present **LISA**, a smart-contract auditor outperforming static analyzers on logic flaws, while Kevin and Yugopuspito (2025) introduce **SmartLLM**, boosting Solidity vulnerability detection. Hybrid and conversational systems Ma et al. (2024); Xia et al. (2024) enhance explainability and exploit reproduction. In healthcare, Neupane et al. (2025) propose a HIPAA-compliant agent framework with PHI sanitization and immutable audit trails. Asthana et al. (2025) develop **OneShield**, a multilingual privacy-guardrail system for PII/PHI detection and OSS risk flagging. For embodied systems, Xing et al. (2025) expose threats from adversarial prompts, sensor spoofing, and instruction misuse, noting that runtime validation reduces but cannot eliminate physical safety risks.

3 Threats to Agentic Systems

The transition from standalone LLMs to autonomous agents introduces a more severe set of

security challenges (Saha et al., 2025,?), as the safety alignment (refusal training) of a base LLM does not reliably transfer to the agentic context (Kumar et al., 2025) . In this section we discuss the threat landscape targeting agentic systems and the frameworks used to evaluate their resilience.

3.1 Attack Surface

3.1.1 Injection Attacks

Prompt injection attacks embed malicious instructions within the prompt fed to an LLM to manipulate it into performing unintended actions (Liu et al., 2024c,a; Yi et al., 2025; Shao et al., 2024). Wang et al. (2025e) identify that the static and predictable structure of an agent's system prompt is a key vulnerability that enables prompt injection attacks to agentic systems. Debenedetti et al. (2024) introduce a benchmark comprising of 97 realistic tasks (e.g., email management, online banking) which reveals a fundamental trade-off: security defenses that reduce vulnerability also degrade the agent's task-completion utility. Liu et al. (2024b) propose split-payload injection attack and find 31 LLM-integrated applications to be vulnerable, including Notion. Several studies show the vulnerability of LLM agents to indirect prompt injection attacks (Zhan et al., 2024; Li et al., 2025a; Yi et al., 2025).

Lee and Tiwari (2024) develop a novel prompt injection attack where a malicious prompt self-replicates across interconnected agents in a multi-agent system like a computer virus and causes system-wide disruption. Dong et al. (2025) propose a memory injection attack that uses crafted prompts to indirectly poison an agent's long-term memory for later malicious execution. Alizadeh et al. (2025) demonstrate that such attacks can cause tool-calling agents to leak sensitive personal data observed during their tasks. Wang et al. (2025f) develop a black-box fuzzing technique that uses Monte Carlo Tree Search to automatically discover indirect prompt injection vulnerabilities by iteratively mutating prompts and environmental observations. Zhang et al. (2025a) and Andriushchenko et al. (2025) design benchmarks that reveal high vulnerability of LLM agents to prompt injection attacks. Zhan et al. (2025) systematically evaluate eight different defenses for LLM agents and demonstrate that all of them can be successfully bypassed by crafting adaptive attacks using established jailbreaking techniques such as GCG (Zou et al., 2023) and AutoDAN (Liu et al., 2024a).

3.1.2 Poisoning and Extraction Attacks

Poisoning attacks present another critical vulnerability for LLM agents by corrupting their memory or knowledge retrieval systems. Fendley et al. (2025) categorize these attacks by their specifications (poison set, trigger, poison behavior, deployment) and define key metrics for evaluation (success rate, stealthiness, persistence). Dong et al. (2025) demonstrate a practical attack that poisons an agent's memory through seemingly benign queries, causing it to execute malicious actions when the poisoned memory is later retrieved by a victim. Similarly, Chen et al. (2024) develop AgentPoison, which poisons an agent's memory or knowledge base by optimizing a backdoor trigger that forces the retrieval of malicious records to hijack its behavior. Zhang et al. (2025a) provide a comprehensive framework for measuring agent vulnerabilities to various attacks, including data poisoning. Several benchmarks (Fu et al., 2025b; Bowen et al., 2025) reveal that larger models do not gain resilience and may even be more susceptible to data poisoning.

Guo et al. (2025b) show that adversaries can repeatedly query an agent's API to obtain a large set of input-output pairs, which can then be used to train an unauthorized "clone" or derivative model, effectively stealing the intellectual property and competitive advantage of the original model provider. These types of attacks are called **model extraction attacks**.

3.1.3 Jailbreak Attacks

Jailbreak attacks attempt to bypass a model's built-in safety measures to force it to produce harmful or unintended content (Wei et al., 2023; Zou et al., 2023; Xu et al., 2024; Lin et al., 2025a). Kumar et al. (2025) and Chiang et al. (2025) both demonstrate that AI agents are significantly more vulnerable to jailbreak attacks than their underlying LLMs. Kumar et al. (2025) and Andriushchenko et al. (2025) show that simple jailbreaking techniques designed for chatbots are highly effective against browser agents, while Chiang et al. (2025) identify three critical design factors (embedding goal directly into system prompt, iterative action generation, and processing environment feedback through event stream) that increase an agent's susceptibility. Andriushchenko et al. (2025) discover that leading LLMs are surprisingly compliant with malicious

agent requests even without jailbreaking. Saha et al. (2025) find that LLM coding agents are highly vulnerable to jailbreak attacks that produce executable malicious code, with attack success rates reaching 75% in multi-file codebases. Yu et al. (2024) use fuzzing techniques to automatically generate novel jailbreak prompts from human-written seeds. Anil et al. (2024) demonstrate a many-shot jailbreak where numerous in-context examples of harmful question answering override a model's safety training. Robey et al. (2024) present a comprehensive exploration of jailbreak attacks on robotic systems powered by LLM agents.

3.1.4 Agent Manipulation Attacks

This class of attacks targets the higher-level cognitive functions of the agent: its planning, reasoning, and goal-setting modules. **Goal hijacking attacks** subtly or overtly alter an agent's objectives, causing it to subvert its original goal (e.g., summarizing a document) to include a secondary, malicious goal (e.g., including advertisements) defined by the attacker (Perez and Ribeiro, 2022; Guo et al., 2025b). Pham and Le (2025) introduce a black-box algorithm that automatically generates malicious system prompts to hijack an LLM's behavior for specific targeted questions, while Chen and Yao (2024) leverage an LLM's weakness in role identification to trick the model into executing a new, malicious task instead of the original one.

Zhang et al. (2025b) introduce an **action hijacking attack** where an agent is tricked into assembling seemingly information data from its own knowledge base into harmful instructions, bypassing input filters. Another class of hijacking attacks is **reward hacking**, which exploits the reward mechanisms in RL-trained agents (Skalse et al., 2022; Pan et al., 2021; Miao et al., 2024; Fu et al., 2025a). These can be caused by reward misgeneralization where models learn from spurious features (Miao et al., 2024), or by agents exploiting reward model ambiguities to maximize their score without true alignment (Fu et al., 2025a). Bondarenko et al. (2025) demonstrate **specification gaming** vulnerabilities, where a capable LLM agent (e.g. OpenAI's o3) instructed to "win against a strong chess engine" hacks the game's environment to ensure victory rather than play fairly, thus satisfying the literal instruction while violating user intent. Finally, a novel threat on multi-agent systems is the presence of a **Byzantine agent**, which is a single compromised or malicious agent that can disrupt the collective's ability to complete a task securely and correctly (Li et al., 2024; Jo and Park, 2025).

3.1.5 Red-Teaming Attacks

Perez et al. (2022) first showed that is is possible to use one LLM to automatically generate test-cases that uncovered harmful behaviors like offensive content and data leakage in a target model. Ge et al. (2023) elevated this to an multi-round iterative setting. He et al. (2025a) develop a directed greybox fuzzing framework designed specifically for detecting taint-style vulnerabilities (such as code injection) in LLM agents. Liu et al. (2025a) introduce the "Agent-in-the-Middle" attack, where an adversarial agent red-teams a system by intercepting and manipulating inter-agent communications. Zhang and Yang (2025) present a search-based framework that simulates multi-turn interactions where an LLM optimizer adversarially co-evolves the strategies of both attacking and defending agents to discover emergent risks.

3.2 Evaluation Frameworks

In this section we discuss broad, multi-faceted benchmarks that are designed to assess multi-faceted vulnerabilities of agentic systems, as well as works that design and analyze specialized environments for the purpose of testing agent security.

3.2.1 Adversarial Benchmarking

Zhang et al. (2025a) introduce ASB benchmark with 10 scenarios and 27 attack classes. RASEval (Fu et al., 2025c) contains 80 attack scenarios in domains like healthcare and finance, demonstrating a 36.8% reduction in task completion under attack. AgentDojo (Debenedetti et al., 2024) uses 97 realistic tasks to highlight the fundamental trade-off between an agent's security and its task-completion utility, while AgentHarm (Andriushchenko et al., 2025) uses a dataset of 110 unique harmful tasks to reveal significant gaps in agent safety alignment. For web agents, SafeArena (Tur et al., 2025) measures completion rates on 250 malicious requests, finding agents complete 34.7% of them, while ST-WebAgentBench (Levy et al., 2025) introduces metrics for policy-compliant success, finding it is 38% lower than standard task completion. For code agents, JAWS-BENCH (Saha et al., 2025) finds up to 75% attack success rates in multi-file codebases, while SandboxEval (Rabin et al., 2025) assesses the security of the execution environment itself with 51 test-cases. InjecAgent (Zhan et al., 2024) offers a

dedicated benchmark for indirect prompt injection attcaks, while BrowserART (Kumar et al., 2025) focuses on the susceptibility to jailbreaks.

3.2.2 Execution Environments

Zhu et al. (2025c) design a sandbox framework that enables LLM agents to interact with exploit vulnerable web applications. Debenedetti et al. (2024) provide a stateful environment with 97 realistic tasks to evaluate the robustness of LLM agents against prompt injection attacks. DoomArena (Boisvert et al., 2025) is a modular red-teaming platform for LLM agents that allows researchers to compose sequential attacks and to mix-and-match adaptive adversary strategies. Zhou et al. (2024) introduce a realistic web environment with 812 long-horizon tasks, where even best performing agents achieve less than 15% success rate.

4 Defense: Hardening the Agents

4.1 Defense & Operations

4.1.1 Secure-by-Design

Recent works (Debenedetti et al., 2024; Li et al., 2025b; Rosario et al., 2025) advance modular and plan–execute isolation, cutting cross-context injection rates by over 40%. Task-level alignment and polymorphic prompting (Jia et al., 2025; Debenedetti et al., 2025; Wang et al., 2025e) employ intent validation and adaptive obfuscation to resist evolving attacks. Governance-oriented frameworks (He et al., 2024; Narajala and Narayan, 2025; Raza et al., 2025; Adabara et al., 2025) extend Secure-by-Design principles through TRiSM-based trust calibration and layered threat modeling. Tang et al. (2024) introduce ModelGuard, constraining knowledge leakage via information-theoretic entropy bounds.

4.1.2 Multi-Agent Security

Secure multi-agent paradigms (Udeshi et al., 2025; Liu et al., 2025b) apply zero-trust and dynamic collaboration to minimize leakage under adversarial conditions. Core vulnerabilities—spoofing, trust delegation, and collusion—are detailed in Han et al. (2025); Ko et al. (2025), motivating formal cross-agent verification. Debate-based collectives (HU et al., 2025; Li et al., 2025d) achieve over 90% phishing detection via randomized smoothing and adversarial consensus, while Lee and Tiwari (2024) uncover LLM-to-LLM prompt infection, highlighting provenance tracking for containment.

4.1.3 Runtime Protection

Reasoning- and knowledge-enhanced guardrails such as R^2-Guard, AgentGuard, and AGrail (Kang and Li, 2024; Xiang et al., 2025; Chen and Cong, 2025; Luo et al., 2025) reduce jailbreak failures by up to 35%. Adaptive systems like PSG-Agent (Wu et al., 2025a) sustain accuracy under evolving threats via personality awareness and continual learning. Deployment studies (Rad et al., 2025; Amazon Web Services, 2024) optimize latency and integrate layered safeguards in production ecosystems. Human-in-the-loop oversight (Wang et al., 2025a) embeds runtime policy enforcement and approval gates for accountability. Behavioral anomaly detectors such as Confront and SentinelAgent (Song et al., 2025; He et al., 2025b) leverage log and graph reasoning for interpretable detection.

4.1.4 Security Operations

Formal verification systems (Kouvaros et al., 2019; Crouse et al., 2024; Lee et al., 2025a; Chen and Cong, 2025) ensure behavioral correctness and runtime assurance through VeriPlan and AgentGuard. LLM-driven analyzers (Yang et al., 2025a; Li et al., 2025e) achieve over 92% accuracy in static analysis, while verification-driven pipelines such as Chain-of-Agents and RepoAudit (Li et al., 2025c; Guo et al., 2025a) operationalize formal assurance. Autonomous response pipelines (Tellache et al., 2025; Molleti et al., 2024) fuse LLM reasoning with threat intelligence, reducing MTTD by 30%. Collaborative frameworks (Liu, 2025; Lin et al., 2025b) like AutoBnB and IRCopilot coordinate triage and remediation. SOC studies (Singh et al., 2025; Wei et al., 2025; Deason et al., 2025) reveal hybrid agent models (e.g., CORTEX) that improve alert precision and reduce fatigue. Rule- and provenance-based threat hunters (Mukherjee and Kantarcioglu, 2025; Schwartz et al., 2025; Meng et al., 2025b; Wu et al., 2025b; Meng et al., 2025a) such as ProvSEEK, LLMCloudHunter, CyberTeam, and ExCyTIn-Bench enable explainable detection and blue-team benchmarking. Cloud-native forensic systems (Alharthi and Yasaei, 2025; Alharthi and Garcia, 2025; Fumero et al., 2025; Tian et al., 2025) like LLM-Powered Forensics, CIAF, CyberSleuth, and GALA automate evidence extraction, reduce triage time by 40%, and improve causal reconstruction.

4.2 Evaluation Frameworks

4.2.1 Benchmarking Platforms

Core testbeds such as AgentDojo, τ-Bench, and TurkingBench (Debenedetti et al., 2024; Yao et al., 2024; Xu et al., 2025b) simulate real-world tasks to evaluate robustness and failure modes of tool-using LLM agents. Safety-focused suites like SafeArena, ST-WebAgentBench, and RAS-Eval (Tur et al., 2025; Levy et al., 2025; Fu et al., 2025c) measure reliability under adversarial stress, while attack-driven frameworks—ASB, AgentHarm, and CVE-Bench (Zhang et al., 2025a; Andriushchenko et al., 2025; Zhu et al., 2025c)—quantify exploitability and vulnerability reproduction. Sandboxed environments such as DoomArena, ToolFuzz, and WebArena (Boisvert et al., 2025; Rabin et al., 2025; Milev et al., 2025; Zhou et al., 2024) further enhance reproducibility, and **aiXamine** (Deniz et al., 2025) offers a streamlined, modular suite for accessible LLM safety evaluation.

4.2.2 Defense Testing

Adaptive studies (Zhan et al., 2025; de Witt, 2025) expose defense fragility under evolving adversaries, urging continuous red-teaming. Broader surveys (Yu et al., 2025; Gan et al., 2024; Deng et al., 2024c) consolidate evolving countermeasures, while Ma et al. (2025); Wang et al. (2025b) emphasize scalable assurance spanning system and governance layers.

4.2.3 Domain-Specific Frameworks

Agentic frameworks in healthcare increasingly embed native defenses against data leakage and policy non-compliance. Shehab (2025) propose *Agentic-AI Healthcare*, a multilingual, privacy-first system using the Model Context Protocol (MCP). Its "Privacy and Compliance Layer" enforces RBAC, AES-GCM field-level encryption, and tamper-evident audit logging, aligning with HIPAA, PIPEDA, and PHIPA standards; embedding compliance structurally rather than adding it post hoc. Beyond healthcare, Wang et al. (2025d) present *PrivacyChecker* and *PrivacyLens-Live* for multi-agent LLM environments. These model-agnostic tools use contextual-integrity reasoning and real-time monitoring to mitigate privacy risks dynamically.In legal domains, Watson et al. (2024) introduce *LAW (Legal Agentic Workflows)*, which reduces hallucinations and clause omissions through tool orchestration and task partitioning. Mhia-Alddin and Hussein (2025) add a contract security layer for expiry tracking, compliance validation, and anomaly detection.

5 Cross-Cutting Analysis and Trends

A cross-cutting analysis of the 151 papers reveals clear structural patterns, detailed in Appendix C.

Architecture. Planner–executor architectures (39.8%) and hybrid models (14%) are now prevalent, replacing earlier monolithic designs (24.6%), which relied on a single LLM for end-to-end reasoning and action.

Agent role. Executor and planner roles dominate (129 papers), with critics/verifiers (95), tool-callers (42), and governors/mediators (24) remaining less common, suggesting a focus on task decomposition, control, and operational efficiency.

LLM backbones. GPT-family models feature in 83% of studies, followed by Claude (71) and LLaMA (63). The rise of open-weight models reflects a push for transparency and reproducibility, but differences in alignment and fine-tuning create fragmentation, hindering cross-model generalization and making consistent safety evaluations challenging.

Knowledge source. Static pre-trained knowledge dominates (132 papers), with adaptive methods like RAG and fine-tuning being limited, suggesting a preference for lightweight, practical deployment. However, this approach may pose security risks in dynamic or evolving threat environments, limiting long-term robustness.

6 Conclusion

This survey explores agentic security in large language model agents and multi-agent systems, highlighting their benefits for cybersecurity and associated risks like prompt injection and agent manipulation. It introduces a taxonomy to classify agentic security based on different paradigms. While defenses like guardrails and architectural hardening exist, they are still fragmented. Future work should focus on benchmarks, cross-domain security, provable safety, scalable governance, and economic impacts. Ensuring trust in these systems is critical as they integrate into key infrastructure.

Limitations

This survey has a few key limitations. It mainly focuses on software-based threats and does not explore physical-world or embodied agent attacks (like those involving robots or sensors) in detail.

Our coverage is also limited to English-language and mostly academic papers, so it may miss industrial or non-English research. In addition, many of the benchmarks we reviewed use synthetic or simplified test setups, which makes it hard to fully judge how well agents would perform in real-world environments. Finally, most studies emphasize accuracy and safety rather than practical aspects like cost, speed, or energy use, and our own taxonomy involves some subjective choices.

References

Ibrahim Adabara, Bashir Olaniyi Sadiq, Aliyu Nuhu Shuaibu, Yale Ibrahim Danjuma, and Venkateswarlu Maninti. 2025. Trustworthy agentic ai systems: A cross-layer review of architectures, threat models, and governance strategies for real-world deployment. *F1000Research*, 14:905.

Dalal Alharthi and Ivan Roberto Kawaminami Garcia. 2025. Cloud investigation automation framework (ciaf): An ai-driven approach to cloud forensics. *Preprint*, arXiv:2510.00452.

Dalal Alharthi and Rozhin Yasaei. 2025. Llm-powered automated cloud forensics: From log analysis to investigation. In *2025 IEEE 18th International Conference on Cloud Computing (CLOUD)*, pages 12–22.

Meysam Alizadeh, Zeynab Samei, Daria Stetsenko, and Fabrizio Gilardi. 2025. Simple prompt injection attacks can leak personal data observed by llm agents during task execution. *Preprint*, arXiv:2506.01055.

Amazon Web Services. 2024. Securing amazon bedrock agents: Safeguarding against indirect prompt injections. AWS Technical Documentation / White Paper. Listed as "LLM AGENT (agent safety orchestrator)".

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, J Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. 2025. Agentharm: A benchmark for measuring harmfulness of LLM agents. In *The Thirteenth International Conference on Learning Representations*.

Cem Anil, Esin DURMUS, Nina Rimsky, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel J Ford, Francesco Mosconi, Rajashree Agrawal, Rylan Schaeffer, Naomi Bashkansky, Samuel Svenningsen, Mike Lambert, Ansh Radhakrishnan, Carson Denison, Evan J Hubinger, and 15 others. 2024. Many-shot jailbreaking. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Mohsen Seyedkazemi Ardebili and Andrea Bartolini. 2025. Kubeintellect: A modular llm-orchestrated agent framework for end-to-end kubernetes management.

Shubhi Asthana, Bing Zhang, Ruchi Mahindru, Chad DeLuca, Anna Lisa Gentile, and Sandeep Gopisetty. 2025. Deploying privacy guardrails for llms: A comparative analysis of real-world applications.

Léo Boisvert, Abhay Puri, Gabriel Huang, Mihir Bansal, Chandra Kiran Reddy Evuru, Avinandan Bose, Maryam Fazel, Quentin Cappart, Alexandre Lacoste, Alexandre Drouin, and Krishnamurthy Dj Dvijotham. 2025. Doomarena: A framework for testing AI agents against evolving security threats. In *Second Conference on Language Modeling*.

Alexander Bondarenko, Denis Volk, Dmitrii Volkov, and Jeffrey Ladish. 2025. Demonstrating specification gaming in reasoning models. *Preprint*, arXiv:2502.13295.

Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. 2025. RepairAgent: An Autonomous, LLM-Based Agent for Program Repair . In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 2188–2200, Los Alamitos, CA, USA. IEEE Computer Society.

Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. 2025. Scaling trends for data poisoning in llms. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26):27206–27214.

Jizhou Chen and Samuel Lee Cong. 2025. Agentguard: Repurposing agentic orchestrator for safety evaluation of tool orchestration. *Preprint*, arXiv:2502.09809.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. 2024. Agentpoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zheng Chen and Buhui Yao. 2024. Pseudo-conversation injection for llm goal hijacking. *Preprint*, arXiv:2410.23678.

Jeffrey Yang Fan Chiang, Seungjae Lee, Jia-Bin Huang, Furong Huang, and Yizheng Chen. 2025. Why are web ai agents more vulnerable than standalone llms? a security analysis. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Maxwell Crouse, Ibrahim Abdelaziz, Ramon Astudillo, Kinjal Basu, Soham Dan, Sadhana Kumaravel, Achille Fokoue, Pavan Kapanipathi, Salim Roukos, and Luis Lastras. 2024. Formally specifying the high-level behavior of llm-based agents. *Preprint*, arXiv:2310.08535.

Isaac David and Arthur Gervais. 2025. Multi-agent penetration testing ai for the web (mapta).

Christian Schroeder de Witt. 2025. Open challenges in multi-agent security: Towards secure systems of interacting ai agents. *Preprint*, arXiv:2505.02077.

Lauren Deason, Adam Bali, Ciprian Bejean, Diana Bolocan, James Crnkovich, Ioana Croitoru, Krishna Durai, Chase Midler, Calin Miron, David Molnar, Brad Moon, Bruno Ostarcevic, Alberto Peltea, Matt Rosenberg, Catalin Sandu, Arthur Saputkin, Sagar Shah, Daniel Stan, Ernest Szocs, and 4 others. 2025. Cybersoceval: Benchmarking llms capabilities for malware analysis and threat intelligence reasoning. *Preprint*, arXiv:2509.20166.

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. *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. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. 2024a. PentestGPT: Evaluating and harnessing large language models for automated penetration testing. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 847–864, Philadelphia, PA. USENIX Association.

Yinlin Deng, Chunqiu Steven Xia, Haoran Peng, Chenyuan Yang, and Lingming Zhang. 2023. Large language models are zero-shot fuzzers: Fuzzing deep-learning libraries via large language models. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2023, page 423–435, New York, NY, USA. Association for Computing Machinery.

Yinlin Deng, Chunqiu Steven Xia, Chenyuan Yang, Shizhuo Dylan Zhang, Shujing Yang, and Lingming Zhang. 2024b. Large language models are edge-case generators: Crafting unusual programs for fuzzing deep learning libraries. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA. Association for Computing Machinery.

Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. 2024c. Ai agents under threat: A survey of key security challenges and future pathways. *Preprint*, arXiv:2406.02630.

Fatih Deniz, Dorde Popovic, Yazan Boshmaf, Euisuh Jeong, Minhaj Ahmad, Sanjay Chawla, and Issa Khalil. 2025. aixamine: Simplified llm safety and security. *Preprint*, arXiv:2504.14985.

Alaeddine Diaf, Abdelaziz Amara Korba, Nour Elislem Karabadji, and Yacine Ghamri-Doudane. 2025. Bart-predict: Empowering iot security with llm-driven cyber threat prediction.

Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen Xiang. 2025. A practical memory injection attack against llm agents. *Preprint*, arXiv:2503.03704.

Mohamad Fakih, Rahul Dharmaji, Halima Bouzidi, Gustavo Quiros Araya, Oluwatosin Ogundare, and Mohammad Abdullah Al Faruque. 2025. Llm4cve: Enabling iterative automated vulnerability repair with large language models. *Preprint*, arXiv:2501.03446.

Richard Fang, Rohan Bindu, Akul Gupta, and Daniel Kang. 2024. Llm agents can autonomously exploit one-day vulnerabilities. *Preprint*, arXiv:2404.08144.

Neil Fendley, Edward W. Staley, Joshua Carney, William Redman, Marie Chau, and Nathan Drenkow. 2025. A systematic review of poisoning attacks against large language models. *Preprint*, arXiv:2506.06518.

Jiayi Fu, Xuandong Zhao, Chengyuan Yao, Heng Wang, Qi Han, and Yanghua Xiao. 2025a. Reward shaping to mitigate reward hacking in rlhf. *Preprint*, arXiv:2502.18770.

Tingchen Fu, Mrinank Sharma, Philip Torr, Shay B Cohen, David Krueger, and Fazl Barez. 2025b. Poisonbench: Assessing large language model vulnerability to poisoned preference data. In *Forty-second International Conference on Machine Learning*.

Yuchuan Fu, Xiaohan Yuan, and Dongxia Wang. 2025c. Ras-eval: A comprehensive benchmark for security evaluation of llm agents in real-world environments. *Preprint*, arXiv:2506.15253.

Stefano Fumero, Kai Huang, Matteo Boffa, Danilo Giordano, Marco Mellia, Zied Ben Houidi, and Dario Rossi. 2025. Cybersleuth: Autonomous blue-team llm agent for web attack forensics. *Preprint*, arXiv:2508.20643.

Yuyou Gan, Yong Yang, Zhe Ma, Ping He, Rui Zeng, Yiming Wang, Qingming Li, Chunyi Zhou, Songze Li, Ting Wang, Yunjun Gao, Yingcai Wu, and Shouling Ji. 2024. Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents. *Preprint*, arXiv:2411.09523.

Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. 2023. Mart: Improving llm safety with multi-round automatic red-teaming. *Preprint*, arXiv:2311.07689.

Luca Gioacchini, Marco Mellia, Idilio Drago, Alexander Delsanto, Giuseppe Siracusano, and Roberto Bifulco. 2024. Autopenbench: Benchmarking generative agents for penetration testing. *Preprint*, arXiv:2410.03225.

Jinyao Guo, Chengpeng Wang, Xiangzhe Xu, Zian Su, and Xiangyu Zhang. 2025a. Repoaudit: An autonomous LLM-agent for repository-level code auditing. In *Forty-second International Conference on Machine Learning*.

Qiming Guo, Jinwen Tang, and Xingran Huang. 2025b. Attacking llms and ai agents: Advertisement embedding attacks against large language models. *Preprint*, arXiv:2508.17674.

Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, and Zhaozhuo Xu. 2025. Llm multi-agent systems: Challenges and open problems. *Preprint*, arXiv:2402.03578.

Andreas Happe and Jürgen Cito. 2025a. Can llms hack enterprise networks? autonomous assumed breach penetration-testing active directory networks. *ACM Trans. Softw. Eng. Methodol.* Just Accepted.

Andreas Happe and Jürgen Cito. 2025b. On the surprising efficacy of llms for penetration-testing. *Preprint*, arXiv:2507.00829.

Pengfei He, Yuping Lin, Shen Dong, Han Xu, Yue Xing, and Hui Liu. 2025a. Red-teaming LLM multi-agent systems via communication attacks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6726–6747, Vienna, Austria. Association for Computational Linguistics.

Xu He, Di Wu, Yan Zhai, and Kun Sun. 2025b. Sentinelagent: Graph-based anomaly detection in multi-agent systems. *Preprint*, arXiv:2505.24201.

Yifeng He, Ethan Wang, Yuyang Rong, Zifei Cheng, and Hao Chen. 2024. Security of ai agents. *Preprint*, arXiv:2406.08689.

Julius Henke. 2025. Autopentest: Enhancing vulnerability management with autonomous llm agents. *Preprint*, arXiv:2505.10321.

Jinwei HU, Yi DONG, Zhengtao DING, and Xiaowei HUANG. 2025. Enhancing robustness of llm-driven multi-agent systems through randomized smoothing. *Chinese Journal of Aeronautics*, page 103779.

Xueyu Hu, Tao Xiong, Biao Yi, Zishu Wei, Ruixuan Xiao, Yurun Chen, Jiasheng Ye, Meiling Tao, Xiangxin Zhou, Ziyu Zhao, Yuhuai Li, Shengze Xu, Shenzhi Wang, Xinchen Xu, Shuofei Qiao, Zhaokai Wang, Kun Kuang, Tieyong Zeng, Liang Wang, and 10 others. 2025. Os agents: A survey on mllm-based agents for computer, phone and browser use.

Isamu Isozaki, Manil Shrestha, Rick Console, and Edward Kim. 2024. Towards automated penetration testing: Introducing llm benchmark, analysis, and improvements. *Preprint*, arXiv:2410.17141.

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, Vienna, Austria. Association for Computational Linguistics.

Yongrae Jo and Chanik Park. 2025. Byzantine-robust decentralized coordination of llm agents. *Preprint*, arXiv:2507.14928.

Mintong Kang and Bo Li. 2024. r^2-guard: Robust reasoning enabled llm guardrail via knowledge-enhanced logical reasoning.

Jan Keller and Jan Nowakowski. 2024. Ai-powered patching: the future of automated vulnerability fixes. Technical report.

Jun Kevin and Pujianto Yugopuspito. 2025. Smartllm: Smart contract auditing using custom generative ai.

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

Ronny Ko, Jiseong Jeong, Shuyuan Zheng, Chuan Xiao, Tae-Wan Kim, Makoto Onizuka, and Won-Yong Shin. 2025. Seven security challenges that must be solved in cross-domain multi-agent llm systems. *Preprint*, arXiv:2505.23847.

Dezhang Kong, Shi Lin, Zhenhua Xu, Zhebo Wang, Minghao Li, Yufeng Li, Yilun Zhang, Hujin Peng, Zeyang Sha, Yuyuan Li, Changting Lin, Xun Wang, Xuan Liu, Ningyu Zhang, Chaochao Chen, Muhammad Khurram Khan, and Meng Han. 2025a. A survey of llm-driven ai agent communication: Protocols, security risks, and defense countermeasures. *Preprint*, arXiv:2506.19676.

He Kong, Die Hu, Jingguo Ge, Liangxiong Li, Tong Li, and Bingzhen Wu. 2025b. Vulnbot: Autonomous penetration testing for a multi-agent collaborative framework. *Preprint*, arXiv:2501.13411.

Panagiotis Kouvaros, Alessio Lomuscio, Edoardo Pirovano, and Hashan Punchihewa. 2019. Formal verification of open multi-agent systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 179–187, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Priyanshu Kumar, Elaine Lau, Saranya Vijayakumar, Tu Trinh, Elaine T Chang, Vaughn Robinson, Shuyan Zhou, Matt Fredrikson, Sean M. Hendryx, Summer Yue, and Zifan Wang. 2025. Aligned LLMs are not aligned browser agents. In *The Thirteenth International Conference on Learning Representations*.

Christine P. Lee, David Porfirio, Xinyu Jessica Wang, Kevin Chenkai Zhao, and Bilge Mutlu. 2025a. Veriplan: Integrating formal verification and llms into end-user planning. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.

Donghyun Lee and Mo Tiwari. 2024. Prompt infection: Llm-to-llm prompt injection within multi-agent systems. *Preprint*, arXiv:2410.07283.

Hwiwon Lee, Ziqi Zhang, Hanxiao Lu, and Lingming Zhang. 2025b. Sec-bench: Automated benchmarking of llm agents on real-world software security tasks. *Preprint*, arXiv:2506.11791.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. 2025. ST-WebAgentBench: A benchmark for evaluating safety & trustworthiness in web agents. In *ArXiv*. ArXiv:2410.06703.

Ang Li, Yin Zhou, Vethavikashini Chithrra Raghuram, Tom Goldstein, and Micah Goldblum. 2025a. Commercial llm agents are already vulnerable to simple yet dangerous attacks. *Preprint*, arXiv:2502.08586.

Evan Li, Tushin Mallick, Evan Rose, William Robertson, Alina Oprea, and Cristina Nita-Rotaru. 2025b. Ace: A security architecture for llm-integrated app systems. *Preprint*, arXiv:2504.20984.

Simin Li, Jun Guo, Jingqiao Xiu, Ruixiao Xu, Xin Yu, Jiakai Wang, Aishan Liu, Yaodong Yang, and Xianglong Liu. 2024. Byzantine robust cooperative multi-agent reinforcement learning as a bayesian game. In *The Twelfth International Conference on Learning Representations*.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, and 11 others. 2025c. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *Preprint*, arXiv:2508.13167.

Wenhao Li, Selvakumar Manickam, Yung wey Chong, and Shankar Karuppayah. 2025d. Phishdebate: An llm-based multi-agent framework for phishing website detection. *Preprint*, arXiv:2506.15656.

Ziyang Li, Saikat Dutta, and Mayur Naik. 2025e. IRIS: LLM-assisted static analysis for detecting security vulnerabilities. In *The Thirteenth International Conference on Learning Representations*.

Liang Lin, Zhihao Xu, Xuehai Tang, Shi Liu, Biyu Zhou, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2025a. Paper summary attack: Jailbreaking llms through llm safety papers. *Preprint*, arXiv:2507.13474.

Xihuan Lin, Jie Zhang, Gelei Deng, Tianzhe Liu, Xiaolong Liu, Changcai Yang, Tianwei Zhang, Qing Guo, and Riqing Chen. 2025b. Ircopilot: Automated incident response with large language models. *Preprint*, arXiv:2505.20945.

Fengyu Liu, Yuan Zhang, Jiaqi Luo, Jiarun Dai, Tian Chen, Letian Yuan, Zhengmin Yu, Youkun Shi, Ke Li, Chengyuan Zhou, Hao Chen, and Min Yang. 2025a. Make agent defeat agent: automatic detection of taint-style vulnerabilities in llm-based agents. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, USA. USENIX Association.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024a. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2024b. Prompt injection attack against llm-integrated applications. *Preprint*, arXiv:2306.05499.

Yinqiu Liu, Ruichen Zhang, Haoxiang Luo, Yijing Lin, Geng Sun, Dusit Niyato, Hongyang Du, Zehui Xiong, Yonggang Wen, Abbas Jamalipour, Dong In Kim, and Ping Zhang. 2025b. Secure multi-llm agentic ai and agentification for edge general intelligence by zero-trust: A survey. *Preprint*, arXiv:2508.19870.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2024c. Formalizing and benchmarking prompt injection attacks and defenses. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, SEC '24, USA. USENIX Association.

Zefang Liu. 2025. Autobnb: Multi-agent incident response with large language models. In *2025 13th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–6.

Weidi Luo, Shenghong Dai, Xiaogeng Liu, Suman Banerjee, Huan Sun, Muhao Chen, and Chaowei Xiao. 2025. Agrail: A lifelong agent guardrail with effective and adaptive safety detection. *Preprint*, arXiv:2502.11448.

Phung Duc Luong, Le Tran Gia Bao, Nguyen Vu Khai Tam, Dong Huu Nguyen Khoa, Nguyen Huu Quyen, Van-Hau Pham, and Phan The Duy. 2025. xoffense: An ai-driven autonomous penetration testing framework with offensive knowledge-enhanced llms and multi agent systems. *Preprint*, arXiv:2509.13021.

Matteo Lupinacci, Francesco Aurelio Pironti, Francesco Blefari, Francesco Romeo, Luigi Arena, and Angelo Furfaro. 2025. The dark side of llms: Agent-based attacks for complete computer takeover. *Preprint*, arXiv:2507.06850.

Wei Ma, Daoyuan Wu, Yuqiang Sun, Tianwen Wang, Shangqing Liu, Jian Zhang, Yue Xue, and Yang Liu. 2024. Combining fine-tuning and llm-based agents for intuitive smart contract auditing with justifications.

Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, Hengyuan Xu, Yunhao Chen, Yunhao Zhao, Hanxun Huang, Yige Li, Yutao Wu, Jiaming Zhang, Xiang Zheng, Yang Bai, Yiming Li, Zuxuan Wu, Xipeng Qiu, and 29 others. 2025. Safety at scale: A comprehensive survey of large model and agent safety. *Foundations and Trends® in Privacy and Security*, 8(3-4):254–469.

Kai Mei, Xi Zhu, Wujiang Xu, Wenyue Hua, Mingyu Jin, Zelong Li, Shuyuan Xu, Ruosong Ye, Yingqiang Ge, and Yongfeng Zhang. 2025. Aios: Llm agent operating system. In *Conference on Language Modeling*.

Ruijie Meng, Martin Mirchev, Marcel Böhme, and Abhik Roychoudhury. 2024. Large language model guided protocol fuzzing. In *Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS)*.

Yuqiao Meng, Luoxi Tang, Feiyang Yu, Jinyuan Jia, Guanhua Yan, Ping Yang, and Zhaohan Xi. 2025a. Uncovering vulnerabilities of llm-assisted cyber threat intelligence. *Preprint*, arXiv:2509.23573.

Yuqiao Meng, Luoxi Tang, Feiyang Yu, Xi Li, Guanhua Yan, Ping Yang, and Zhaohan Xi. 2025b. Benchmarking llm-assisted blue teaming via standardized threat hunting. *Preprint*, arXiv:2509.23571.

– Mhia-Alddin and – Hussein. 2025. Ai-powered contract security: Managing expiry, compliance, and risk mitigation through deep learning and llms. In *Proceedings / Chapters in AI / Applied Deep Learning Legal Tech*. Chapter / working publication, April 2025.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. 2024. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling.

Ivan Milev, Mislav Balunović, Maximilian Baader, and Martin Vechev. 2025. Toolfuzz – automated agent tool testing. *Preprint*, arXiv:2503.04479.

Ramasankar Molleti, Vinod Goje, Puneet Luthra, and Prathap Raghavan. 2024. Automated threat detection and response using llm agents. *World Journal of Advanced Research and Reviews*, 24:079–090.

Mykyta Mudryi, Markiyan Chaklosh, and Grzegorz Wójcik. 2025. The hidden dangers of browsing ai agents.

Kunal Mukherjee and Murat Kantarcioglu. 2025. Llm-driven provenance forensics for threat investigation and detection. *Preprint*, arXiv:2508.21323.

Lajos Muzsai, David Imolai, and András Lukács. 2024. Hacksynth: Llm agent and evaluation framework for autonomous penetration testing. *Preprint*, arXiv:2412.01778.

Vineeth Sai Narajala and Om Narayan. 2025. Securing agentic ai: A comprehensive threat model and mitigation framework for generative ai agents. *Preprint*, arXiv:2504.19956.

Subash Neupane, Shaswata Mitra, Sudip Mittal, and Shahram Rahimi. 2025. Towards a hipaa compliant agentic ai system in healthcare.

Tomas Nieponice, Veronica Valeros, and Sebastian Garcia. 2025. Aracne: An llm-based autonomous shell pentesting agent. *Preprint*, arXiv:2502.18528.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2021. The effects of reward misspecification: Mapping and mitigating misaligned models. In *Deep RL Workshop NeurIPS 2021*.

Melissa Z Pan, Mert Cemri, Lakshya A Agrawal, Shuyi Yang, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Kannan Ramchandran, Dan Klein, Joseph E. Gonzalez, Matei Zaharia, and Ion Stoica. 2025. Why do multiagent systems fail? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *Preprint*, arXiv:2202.03286.

Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models. *Preprint*, arXiv:2211.09527.

Viet Pham and Thai Le. 2025. Cain: Hijacking llm-humans conversations via malicious system prompts. *Preprint*, arXiv:2505.16888.

Rafiqul Rabin, Jesse Hostetler, Sean McGregor, Brett Weir, and Nick Judd. 2025. Sandboxeval: Towards securing test environment for untrusted code. *Preprint*, arXiv:2504.00018.

Melissa Kazemi Rad, Huy Nghiem, Sahil Wadhwa, Andy Luo, and Mohammad Shahed Sorower. 2025. Refining input guardrails: Enhancing LLM-as-a-judge efficiency through chain-of-thought fine-tuning and alignment. In *AAAI 2025 Workshop on Preventing and Detecting LLM Misinformation (PDLM)*.

Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. 2025. Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems. *Preprint*, arXiv:2506.04133.

Alexander Robey, Zachary Ravichandran, Vijay Kumar, Hamed Hassani, and George J. Pappas. 2024. Jailbreaking llm-controlled robots. *Preprint*, arXiv:2410.13691.

Ron F. Del Rosario, Klaudia Krawiecka, and Christian Schroeder de Witt. 2025. Architecting resilient llm agents: A guide to secure plan-then-execute implementations. *Preprint*, arXiv:2509.08646.

Bikash Saha and Sandeep Kumar Shukla. 2025. Malgen: A generative agent framework for modeling malicious software in cybersecurity. *Preprint*, arXiv:2506.07586.

Shoumik Saha, Jifan Chen, Sam Mayers, Sanjay Krishna Gouda, Zijian Wang, and Varun Kumar. 2025. Breaking the code: Security assessment of ai code agents through systematic jailbreaking attacks. *Preprint*, arXiv:2510.01359.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yuval Schwartz, Lavi Ben-Shimol, Dudu Mimran, Yuval Elovici, and Asaf Shabtai. 2025. Llmcloudhunter: Harnessing llms for automated extraction of detection rules from cloud-based cti. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 1922–1941, New York, NY, USA. Association for Computing Machinery.

Zedian Shao, Hongbin Liu, Jaden Mu, and Neil Zhenqiang Gong. 2024. Enhancing prompt injection attacks to llms via poisoning alignment.

Mohammed A. Shehab. 2025. Agentic-ai healthcare: Multilingual, privacy-first framework with mcp agents. *arXiv preprint*, arXiv:2510.02325. Preprint, submitted 25 Sep 2025, cs.CR / cs.AI.

Xiangmin Shen, Lingzhi Wang, Zhenyuan Li, Yan Chen, Wencheng Zhao, Dawei Sun, Jiashui Wang, and Wei Ruan. 2025. Pentestagent: Incorporating llm agents to automated penetration testing. In *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '25, page 375–391, New York, NY, USA. Association for Computing Machinery.

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

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Brian Singer, Keane Lucas, Lakshmi Adiga, Meghna Jain, Lujo Bauer, and Vyas Sekar. 2025. On the feasibility of using llms to autonomously execute multi-host network attacks. *Preprint*, arXiv:2501.16466.

Ronal Singh, Shahroz Tariq, Fatemeh Jalalvand, Mohan Baruwal Chhetri, Surya Nepal, Cecile Paris, and Martin Lochner. 2025. Llms in the soc: An empirical study of human-ai collaboration in security operations centres. *Preprint*, arXiv:2508.18947.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward hacking. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Shuang Song, Yifei Zhang, and Neng Gao. 2025. Confront insider threat: Precise anomaly detection in behavior logs based on LLM fine-tuning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8589–8601, Abu Dhabi, UAE. Association for Computational Linguistics.

Izaiah Sun, Daniel Tan, and Andy Deng. 2025. Lisa technical report: An agentic framework for smart contract auditing.

Minxue Tang, Anna Dai, Louis DiValentin, Aolin Ding, Amin Hass, Neil Zhenqiang Gong, Yiran Chen, and Hai "Helen" Li. 2024. ModelGuard: Information-Theoretic defense against model extraction attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 5305–5322, Philadelphia, PA. USENIX Association.

Amine Tellache, Abdelaziz Amara Korba, Amdjed Mokhtari, Horea Moldovan, and Yacine Ghamri-Doudane. 2025. Advancing autonomous incident response: Leveraging llms and cyber threat intelligence. *Preprint*, arXiv:2508.10677.

Yifang Tian, Yaming Liu, Zichun Chong, Zihang Huang, and Hans-Arno Jacobsen. 2025. Gala: Can graph-augmented large language model agentic workflows elevate root cause analysis? *Preprint*, arXiv:2508.12472.

Dheer Toprani and Vijay K. Madisetti. 2025. Llm agentic workflow for automated vulnerability detection and remediation in infrastructure-as-code. *IEEE Access*, 13:69175–69181.

Ada Defne Tur, Nicholas Meade, Xing Han Lù, Alejandra Zambrano, Arkil Patel, Esin DURMUS, Spandana Gella, Karolina Stanczak, and Siva Reddy. 2025. Safearena: Evaluating the safety of autonomous web agents. In *Forty-second International Conference on Machine Learning*.

Meet Udeshi, Minghao Shao, Haoran Xi, Nanda Rani, Kimberly Milner, Venkata Sai Charan Putrevu, Brendan Dolan-Gavitt, Sandeep Kumar Shukla, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, and Muhammad Shafique. 2025. D-cipher: Dynamic collaborative intelligent multi-agent system with planner and heterogeneous executors for offensive security. *Preprint*, arXiv:2502.10931.

Saad Ullah, Praneeth Balasubramanian, Wenbo Guo, Amanda Burnett, Hammond Pearce, Christopher Kruegel, Giovanni Vigna, and Gianluca Stringhini. 2025. From cve entries to verifiable exploits: An automated multi-agent framework for reproducing cves. *Preprint*, arXiv:2509.01835.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. 2025a. Agentspec: Customizable runtime enforcement for safe and reliable llm agents. *Preprint*, arXiv:2503.18666.

Junjie Wang, Yuchao Huang, Chunyang Chen, Zhe Liu, Song Wang, and Qing Wang. 2024. Software testing with large language models: Survey, landscape, and vision. 50(4):911–936.

Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, Liang Lin, Zhihao Xu, Haolang Lu, Xinye Cao, Xinyun Zhou, Weifei Jin, Fanci Meng, Shicheng Xu, Junyuan Mao, and 84 others. 2025b. A comprehensive survey in llm(-agent) full stack safety: Data, training and deployment. *Preprint*, arXiv:2504.15585.

Peiran Wang, Xiaogeng Liu, and Chaowei Xiao. 2025c. CVE-bench: Benchmarking LLM-based software engineering agent's ability to repair real-world CVE vulnerabilities. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4207–4224, Albuquerque, New Mexico. Association for Computational Linguistics.

Shouju Wang, Fenglin Yu, Xirui Liu, Xiaoting Qin, Jue Zhang, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. 2025d. Privacy in action: Towards realistic privacy mitigation and evaluation for llm-powered agents. *arXiv preprint*, arXiv:2509.17488. Preprint, submitted 22 Sep 2025, cs.CR / cs.AI.

Zhilong Wang, Neha Nagaraja, Lan Zhang, Hayretdin Bahsi, Pawan Patil, and Peng Liu. 2025e. To protect the llm agent against the prompt injection attack with polymorphic prompt. *Preprint*, arXiv:2506.05739.

Zhun Wang, Vincent Siu, Zhe Ye, Tianneng Shi, Yuzhou Nie, Xuandong Zhao, Chenguang Wang, Wenbo Guo, and Dawn Song. 2025f. Agentvigil: Generic black-box red-teaming for indirect prompt injection against llm agents. *Preprint*, arXiv:2505.05849.

Ziyue Wang and Liyi Zhou. 2025. Agentic discovery and validation of android app vulnerabilities. *Preprint*, arXiv:2508.21579.

William Watson, Nicole Cho, Nishan Srishankar, Zhen Zeng, Lucas Cecchi, Daniel Scott, Suchetha Siddagangappa, Rachneet Kaur, Tucker Balch, and Manuela Veloso. 2024. Law: Legal agentic workflows for custody and fund services contracts. *arXiv preprint*, arXiv:2412.11063. Preprint, submitted 15 Dec 2024, cs.AI.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: how does llm safety training fail? In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Bowen Wei, Yuan Shen Tay, Howard Liu, Jinhao Pan, Kun Luo, Ziwei Zhu, and Chris Jordan. 2025. Cortex: Collaborative llm agents for high-stakes alert triage. *Preprint*, arXiv:2510.00311.

Yaozu Wu, Jizhou Guo, Dongyuan Li, Henry Peng Zou, Wei-Chieh Huang, Yankai Chen, Zhen Wang, Weizhi Zhang, Yangning Li, Meng Zhang, Renhe Jiang, and Philip S. Yu. 2025a. Psg-agent: Personality-aware safety guardrail for llm-based agents. *Preprint*, arXiv:2509.23614.

Yiran Wu, Mauricio Velazco, Andrew Zhao, Manuel Raúl Meléndez Luján, Srisuma Movva, Yogesh K Roy, Quang Nguyen, Roberto Rodriguez, Qingyun Wu, Michael Albada, Julia Kiseleva, and Anand Mudgerikar. 2025b. Excytin-bench: Evaluating llm agents on cyber threat investigation. *Preprint*, arXiv:2507.14201.

Shihao Xia, Shuai Shao, Mengting He, Tingting Yu, Linhai Song, and Yiying Zhang. 2024. Auditgpt: Auditing smart contracts with chatgpt.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. 2025. Guardagent: Safeguard llm agents by a guard agent via knowledge-enabled reasoning. *Preprint*, arXiv:2406.09187.

Wenpeng Xing, Minghao Li, Mohan Li, and Meng Han. 2025. Towards robust and secure embodied ai: A survey on vulnerabilities and attacks.

Hanxiang Xu, Shenao Wang, Ningke Li, Kailong Wang, Yanjie Zhao, Kai Chen, Ting Yu, Yang Liu, and Haoyu Wang. 2025a. Large language models for cyber security: A systematic literature review. *ACM Trans. Softw. Eng. Methodol.*

Kevin Xu, Yeganeh Kordi, Tanay Nayak, Adi Asija, Yizhong Wang, Kate Sanders, Adam Byerly, Jingyu Zhang, Benjamin Van Durme, and Daniel Khashabi. 2025b. TurkingBench: A challenge benchmark for web agents. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3694–3710, Albuquerque, New Mexico. Association for Computational Linguistics.

Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. 2024. A comprehensive study of jailbreak attack versus defense for large language models. *Preprint*, arXiv:2402.13457.

Chenyuan Yang, Zijie Zhao, Zichen Xie, Haoyu Li, and Lingming Zhang. 2025a. Knighter: Transforming static analysis with llm-synthesized checkers. In *Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles*, SOSP '25, New York, NY, USA. Association for Computing Machinery.

Zhenning Yang, Archit Bhatnagar, Yiming Qiu, Tongyuan Miao, Patrick Tser Jern Kon, Yunming Xiao, Yibo Huang, Martin Casado, and Ang Chen. 2025b. Cloud infrastructure management in the age of ai agents. *ACM SIGOPS Operating Systems Review*, 59(1).

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Ziyang Ye, Triet Huynh Minh Le, and M. Ali Babar. 2025. Llmsecconfig: An llm-based approach for fixing software container misconfigurations.

Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. 2025. Benchmarking and defending against indirect prompt injection attacks on large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, page 1809–1820, New York, NY, USA. Association for Computing Machinery.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2024. LLM-Fuzzer: Scaling assessment of large language model jailbreaks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4657–4674, Philadelphia, PA. USENIX Association.

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pan, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, Bo An, and Qingsong Wen. 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*, KDD '25, page 6216–6226, New York, NY, USA. Association for Computing Machinery.

Qiusi Zhan, Richard Fang, Henil Shalin Panchal, and Daniel Kang. 2025. Adaptive attacks break defenses against indirect prompt injection attacks on LLM agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7101–7117, Albuquerque, New Mexico. Association for Computational Linguistics.

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, Bangkok, Thailand. Association for Computational Linguistics.

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 *ICLR*.

Yanzhe Zhang and Diyi Yang. 2025. Searching for privacy risks in llm agents via simulation. *Preprint*, arXiv:2508.10880.

Yuyang Zhang, Kangjie Chen, Jiaxin Gao, Ronghao Cui, Run Wang, Lina Wang, and Tianwei Zhang. 2025b. Towards action hijacking of large language model-based agent. *Preprint*, arXiv:2412.10807.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. *Preprint*, arXiv:2307.13854.

Jie Zhu, Chihao Shen, Ziyang Li, Jiahao Yu, Yizheng Chen, and Kexin Pei. 2025a. Locus: Agentic predicate synthesis for directed fuzzing. *Preprint*, arXiv:2508.21302.

Yuxuan Zhu, Antony Kellermann, Dylan Bowman, Philip Li, Akul Gupta, Adarsh Danda, Richard Fang, Conner Jensen, Eric Ihli, Jason Benn, Jet Geronimo, Avi Dhir, Sudhit Rao, Kaicheng Yu, Twm Stone, and Daniel Kang. 2025b. Cve-bench: A benchmark for ai agents' ability to exploit real-world web application vulnerabilities. *Preprint*, arXiv:2503.17332.

Yuxuan Zhu, Antony Kellermann, Dylan Bowman, Philip Li, Akul Gupta, Adarsh Danda, Richard Fang, Conner Jensen, Eric Ihli, Jason Benn, Jet Geronimo, Avi Dhir, Sudhit Rao, Kaicheng Yu, Twm Stone, and Daniel Kang. 2025c. CVE-bench: A benchmark for AI agents' ability to exploit real-world web application vulnerabilities. In *Forty-second International Conference on Machine Learning*.

Yuxuan Zhu, Antony Kellermann, Akul Gupta, Philip Li, Richard Fang, Rohan Bindu, and Daniel Kang. 2025d. Teams of llm agents can exploit zero-day vulnerabilities. *Preprint*, arXiv:2406.01637.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

A Paper Collection Methodology

To ensure a comprehensive and reproducible review of the agentic security landscape, we employed a multi-stage paper collection methodology combining automated searches, manual curation, and snowballing techniques.

Automated Database Search

We conducted an automated search across major academic repositories—including ACL Anthology, IEEE Xplore, ACM Digital Library, and arXiv—covering publications from January 2023 to September 2025. Using a Boolean query, we combined two groups keywords related to agentic systems and security concepts.

Search Query Structure `(Group 1 Keywords) AND (Group 2 Keywords)`

- **Group 1 (Agent-related):** `("LLM agent" OR "AI agent" OR "agentic AI" OR "autonomous agent" OR "multi-agent system")`
- **Group 2 (Security-related):** `("security" OR "threat" OR "vulnerability" OR "attack" OR "defense" OR "red team" OR "blue team" OR "penetration testing" OR "fuzzing" OR "jailbreak" OR "prompt injection" OR "poisoning" OR "hardening" OR "adversarial")`

Manual Curation

To identify relevant work that our keyword search may had missed, we manually scanned the proceedings of top-tier security (e.g., USENIX Security, ACM CCS, NDSS) and AI (e.g., ACL, EMNLP, NeurIPS, ICLR, ICML) conferences from the same period.

Inclusion and Exclusion Criteria

We applied a strict set of criteria to the collected papers to ensure the relevance and focus of our survey.

Inclusion Criteria The paper's primary subject must be **LLM-based agents**. It must have a substantial focus on a **technical security aspect**, aligning with one of our three pillars. The work must be a peer-reviewed publication or a highly-cited preprint.

Exclusion Criteria We excluded papers on general LLM safety that do not address agentic systems, studies on non-LLM agents, works centered on high-level ethics or policy without technical details, and non-technical articles.

Snowballing

Finally, we performed backward and forward snowballing on the curated set of included papers. We reviewed the reference lists of these key papers to identify foundational or related works we might have missed.

B Related Works and Gap Analysis

Recent surveys have started exploring how large language models (LLMs) and agentic systems intersect, but most of them only cover part of the full picture. He et al. (2024) study concrete system-level vulnerabilities and practical defenses like sandbox settings, session management, and encryption schemes. While they provide useful technical insights, their scope is limited to specific systems. Our work builds on such foundations by connecting these defenses to broader architectural and governance perspectives across multiple domains.

Yu et al. (2025) focus on trustworthiness, discussing issues like safety, privacy, fairness, and robustness. Their survey is strong in analyzing threats and defenses but lacks coverage of real-world applications. In comparison, we take a more direct security-focused approach, showing how these challenges play out in practice through concrete agent behaviors, attack vectors, and defensive mechanisms.

Raza et al. (2025) discuss the TRiSM framework (Trust, Risk, and Security Management) and view agentic AI through enterprise governance and compliance lenses. Their work highlights risk management and oversight but does not explore how security is built into agent design. Our survey complements this by examining technical, architectural, and behavioral aspects that directly influence security in operation.

Deng et al. (2024c) provide a good overview of LLM-agent threats such as prompt injection and data poisoning but do not connect these threats to autonomy levels or coordinated defenses. We extend this line of work by treating agentic security as a layered system, showing how design choices and coordination roles shape overall safety.

Ma et al. (2025) review safety across several model families, including agents, and present many threats, defenses, and benchmarks. However, their coverage of specific agent applications is limited. Our survey focuses on the unique behaviors of autonomous agents, exploring their dynamic decision-making, real-world risks, and operational defenses.

de Witt (2025) discuss multi-agent security conceptually and identify potential risks like collusion or oversight failure. They set an important theoretical base, and we build on it by connecting those ideas to measurable defenses and empirical evaluations in real agent environments.

Li et al. (2025a) show that commercial LLM agents can be easily manipulated in practice, offering valuable real-world evidence of vulnerabilities. We take these findings a step further by situating them in a broader taxonomy that explains why such failures occur and how defenses can be systematically designed.

Kong et al. (2025a) analyze communication protocols and the related security risks in how agents interact with users, other agents, and environments. Their work focuses mainly on communication-level threats, while our survey connects these communication issues to higher-level coordination, control, and defense strategies across full agentic ecosystems.

Wang et al. (2025b) describe safety risks throughout the LLM lifecycle, from data and training to deployment. Their work captures the broader model development pipeline, whereas we focus on what happens after deployment—how agents be-

have, interact, and defend themselves in the real world.

C Cross-Cutting Analysis and Emerging Trends

Our synthesis of 151 papers uncovers distinct structural patterns and maturity gradients in LLM-based agentic security research. The field is consolidating around modular architectures and hybrid reasoning frameworks, yet continues to face foundational gaps in autonomy calibration, coordination trust, and adaptive learning. This section summarizes key architectural, functional, and knowledge-level trends, grounded in quantitative distributions shown in Figures 2–6.

C.1 Architecture and Autonomy

As shown in Figure 2, the field exhibits a pronounced convergence toward the **planner–executor** paradigm (68 papers) with *hybrid architectures* (24) that fuse symbolic reasoning or retrieval modules with neural controllers. These replace earlier monolithic designs (42) relying on a single LLM for end-to-end reasoning and action. This shift reflects a growing appreciation of decomposed cognitive pipelines, where planning, execution, and verification can be modularized to improve interpretability and debugging. In terms of autonomy, more than half of the surveyed systems (**71 papers**) implement bounded automation, allowing agents to act independently within predefined limits, while 46 incorporate human approval gates and 18 restrict themselves to advisory roles.

C.2 Role Stratification

Functional role distributions in Figure 3 confirm the dominance of executor and planner roles (129 and 107 papers, respectively), reflecting the field's emphasis on task decomposition and control. Critics/verifiers appear in 95 papers, marking the rise of introspective subagents. In contrast, *tool-callers* (42) and *governors/mediators* (24) remain sparse but pivotal, typically employed in systems focused on self-regulation, ethical alignment, or conflict resolution. This emerging role ecology signals a slow shift from monolithic reasoning to layered, self-checking collectives.

C.3 LLM Ecosystem Fragmentation and Dependence

As illustrated in Figure 4, **GPT-family models** dominate the landscape (126 papers), establishing de facto benchmark status but raising concerns about monoculture and reproducibility. *Claude* (71) and *LLaMA* (63) constitute the next major clusters, followed by *Gemini* (48), *Mistral* (30), *Qwen* (18), and *DeepSeek* (15). The gradual rise of open-weight backbones reflects a push toward transparent and auditable research, particularly for security-sensitive evaluation. However, model-specific alignment differences create fragmentation: safety fine-tuning and evaluation pipelines are rarely transferable, hindering cross-model generalization and reproducibility.

C.4 Modalities and Data Context Expansion

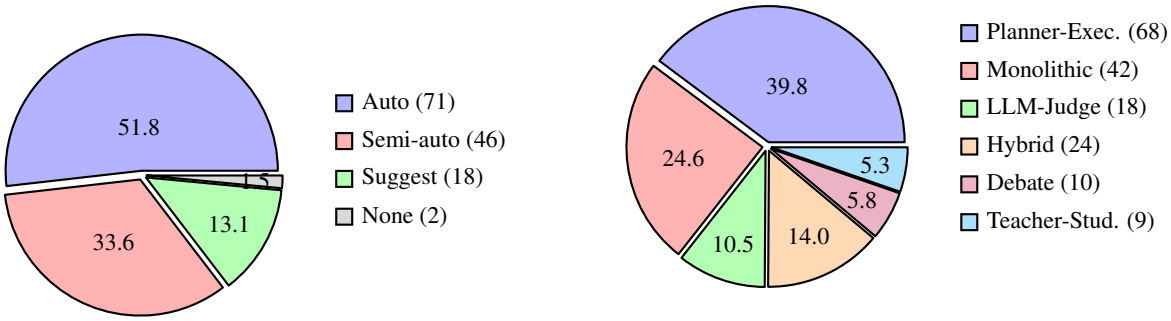
The input modality spectrum (Figure 5) reveals the centrality of text (141 papers), but with notable diversification into structured and operational data sources. Log analysis (101 papers) and code reasoning (93) dominate among non-text modalities, underscoring the dual nature of security agents as both interpreters and executors of programmatic artifacts. Fewer systems operate on network traces (38), images (11), or binaries (10), yet these categories are gaining traction in intrusion detection and reverse-engineering contexts. The integration of such multimodal signals represents a key direction for enabling agents to correlate telemetry, policy, and execution behavior within unified analytic pipelines.

C.5 Knowledge Sourcing and Adaptivity Gaps

Figure 6 highlights the dominance of static pretrained knowledge (132 papers), with limited adoption of adaptive learning paradigms. In-context learning (66) and RAG (43) show partial adoption, while fine-tuning (38) and reinforcement or preference optimization (8) remain niche. This imbalance constrains resilience to evolving threats and undermines reproducibility in adversarial contexts. Reliance on frozen priors suggests a community preference for lightweight deployment over continual learning, practical but potentially insecure in dynamic threat environments. Going forward, secure RAG pipelines with verified provenance, incremental fine-tuning, and model distillation will be key to balancing adaptivity with stability.

D Benchmark Inventory

Table 2 provides a detailed analysis of the existing adversarial benchmarks.



(a) Autonomy Levels

(b) Architecture Patterns

Figure 2: Agent Architecture Distribution across 151 papers

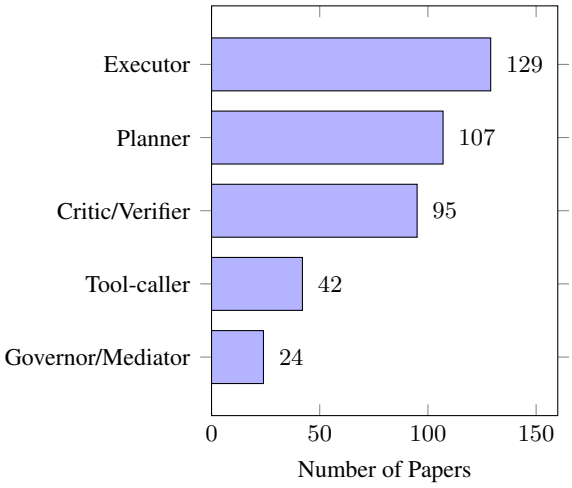


Figure 3: Agent role distribution across 151 papers

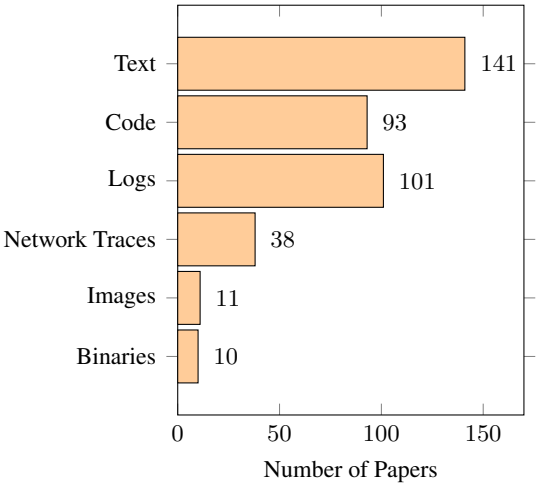


Figure 5: Data modality distribution across 151 papers

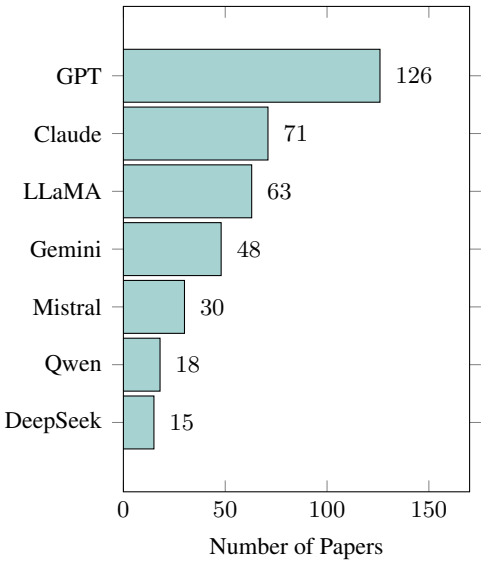


Figure 4: Most tested LLM backbones across 151 papers

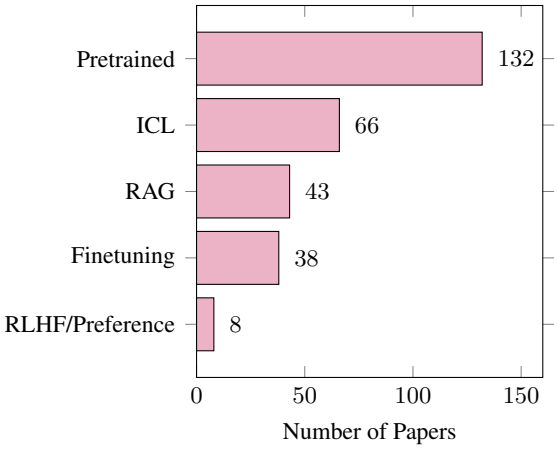


Figure 6: Primary knowledge sources of agentic systems.

Table 2: Adversarial Benchmarks for LLM Agents

Benchmark	Environment	Attacks / Threat Model	Findings	Insights
ASB (Zhang et al., 2025a)	Multi-domain agent tasks with 400+ tools; 10 scenarios; standardized evaluation harness.	Prompt injection **(primary)**, memory attacks, data poisoning, unauthorized tool invocation, privilege escalation; 27 attack/defense classes.	Existing agents highly vulnerable; many fail even simple attack tasks; reports refusal rate and a unified resilience metric.	Standardized, reproducible testbed spanning both offensive and defensive evaluation; clear taxonomy centered on prompt-injection surfaces.
RAS-Eval (Fu et al., 2025c)	Real-world domains (finance, healthcare); 80 scenarios / 3,802 tasks; simulation and real tool use.	11 CWE categories; broad adversarial stress across realistic workflows.	Task completion drops by ∼36.8% on average (up to 85.7%) under attack.	Maps agent failures to CWE; couples domain realism with measurable robustness deltas.
AgentDojo (Debenedetti et al., 2024)	Dynamic, stateful env.; 97 realistic multi-turn tool tasks (e.g., email, banking) with formal, deterministic checks.	Prompt injection via untrusted data/tools; security vs. utility trade-off analysis.	Defenses reduce attack success but degrade task utility; SOTA LLMs struggle on realistic pipelines.	Makes the *security–utility* trade-off explicit; judge is environment-state based (no LLM-as-judge).
AgentHarm (Andriushchenko et al., 2025)	Agent tasks spanning 110 harmful tasks across 11 harm categories.	Jailbreaks, indirect injections, self-compromising actions, unsafe code execution.	Significant gaps in compliance and contextual safety across agents.	Introduces robustness, refusal accuracy, and ethical consistency metrics focused on harm reduction.
SafeArena (Tur et al., 2025)	Web agents across multiple websites; 250 benign vs. 250 harmful tasks.	Malicious requests: misinformation, illegal actions, malware-related behaviors.	SOTA (e.g., GPT–4o) completes 34.7% of malicious requests.	Demonstrates real web-workflow risks; quantifies unsafe completions under realistic browsing.
ST-WebAgentBench (Levy et al., 2025)	Enterprise-like web tasks: 222 tasks with 646 policy instances.	Policy compliance (consent, data boundaries); defines CuP, pCuP, and Risk Ratio.	Policy-compliant success is ≈38% lower than standard completion.	Shifts evaluation beyond raw success to *trust/safety-constrained* success.
JAWS-BENCH (Saha et al., 2025)	Code agents with executable-aware judging across JAWS-0/1/M (empty, single-file, multi-file).	Systematic jailbreaking to elicit harmful, *executable* code; tests compliance, attack success, compile, run.	Up to 75% attack success in multi-file codebases.	Execution-grounded judging prevents false safety from mere textual refusals; highlights multi-file risks.
SandboxEval (Rabin et al., 2025)	Code-execution testbeds; 51 hand-crafted sandbox test cases (applied to Dyff).	Dangerous behaviors: FS tampering, data exfiltration, network access, etc.	Naive sandbox configurations can be compromised by malicious code.	Security must include *runtime isolation posture*, not only agent policy.
BrowserART (Kumar et al., 2025)	Browser-agent red-teaming toolkit across synthetic & real sites (100 harmful behaviors).	Jailbreaks against browser agents; transfer of chatbot jailbreaks to agentic setting.	Backbone LLM refusal *does not* transfer: with human rewrites, GPT–4o pursued 98/100, o1-preview 63/100 harmful behaviors.	Agentic, tool-using context weakens safety adherence even without exotic attacks.
InjecAgent (Zhan et al., 2024)	Tool-integrated agents; 1,054 test cases across 17 user tools and 62 attacker tools.	*Indirect* prompt injections via external content, API outputs, chained tools; path-aware categorization.	Well-aligned agents frequently execute compromised instructions under indirect injections.	Provides fine-grained, propagation-path metrics; standardizes indirect-injection stress for tool-augmented agents.