A SURVEY ON AGENTIC SECURITY: APPLICATIONS, THREATS AND DEFENSES

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work we present the first holistic survey of the agentic security landscape, structuring the field around three fundamental pillars: Applications, Threats, and Defenses. We provide a comprehensive taxonomy of over 160 papers, explaining how agents are used in downstream cybersecurity applications, inherent threats to agentic systems, and countermeasures designed to protect them. A detailed cross-cutting analysis shows emerging trends in agent architecture while revealing critical research gaps in model, modality and benchmark 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. We define an **LLM Agent** as a 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. This newfound agency has enabled LLM-agents to demonstrate remarkable capabilities across the security spectrum (Shen et al., 2025; Zhu et al., 2025c; 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 160 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. We follow a systematic paper collection methodology to ensure coverage and reproducibility, which is described in Appendix A. 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 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 identify key trends and gaps, including migration from monolithic to planner-executor and hybrid agents, backbone LLM monopoly (GPT), uneven threat and modality coverage, and benchmark fragmentation.

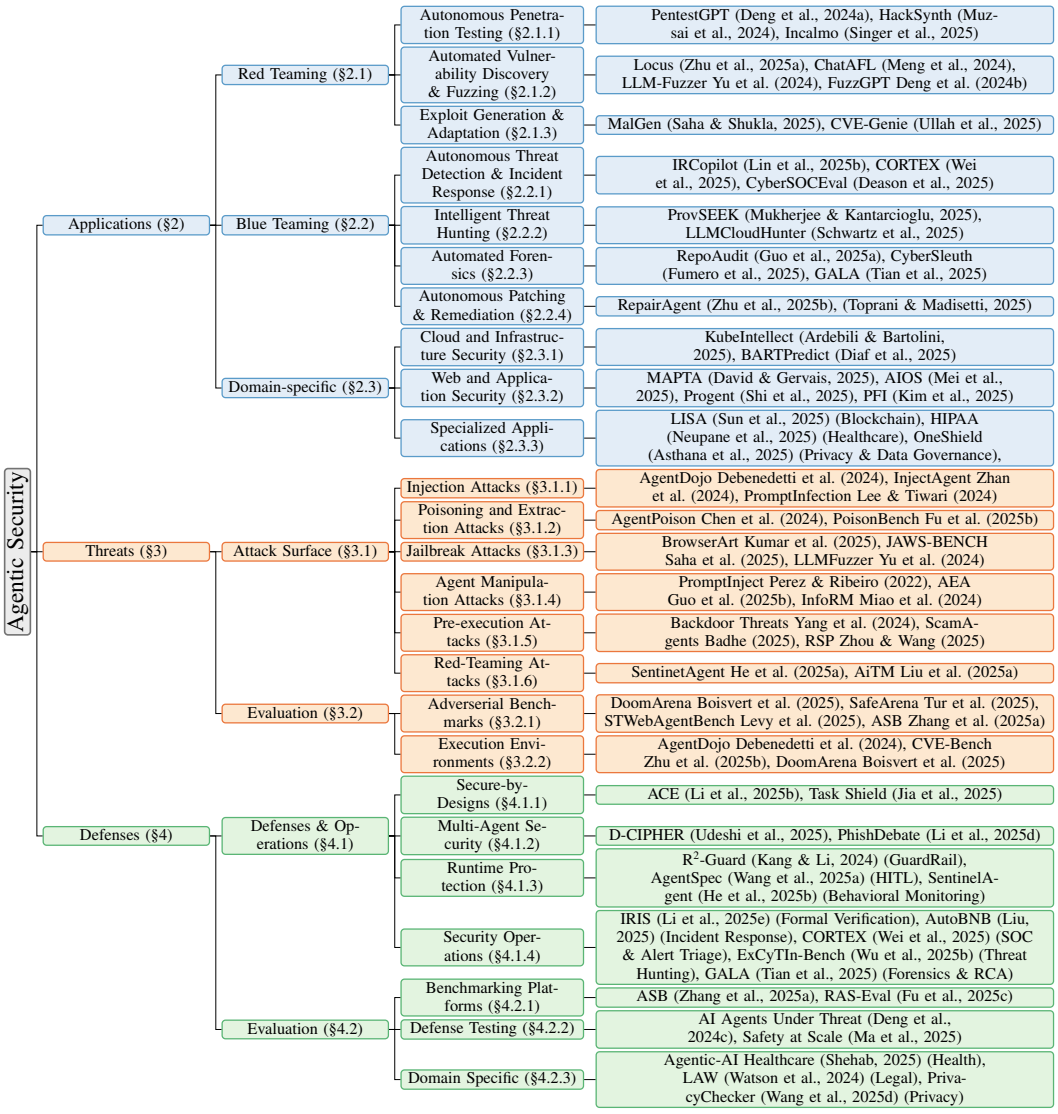


Figure 1: Overview of Agentic Security Taxonomy

2 Applications of Agents in Security

This section describes how agents are applied across the cybersecurity landscape, from offensive testing and exploit generation to defensive detection, forensics, and automated remediation.

2.1 Offensive Security Agents (Red-Teaming)

This subsection describes autonomous and reasoning-driven red-team agentic systems that conduct penetration testing, vulnerability discovery, fuzzing, and exploit adaptation.

Table 1: Survey comparison. Legend: ✓ = covered; △ = partial/limited; ✗ = not covered.

Survey	Applications	Threats	Defenses	Benchmarks	Focus
Yu et al. (2025)	✗	✓	✓	✓	Trustworthiness, robustness, privacy
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)
Wang et al. (2025b)	✗	✓	✗	△	Safety risks during model development pipeline
de Witt (2025)	✗	△	△	✗	Theoretical basis for multi-agent risks
This Survey	✓	✓	✓	✓	**Holistic agentic security**

2.1.1 Autonomous Penetration Testing

These agents perform autonomous end-to-end penetration testing 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 & 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 & 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

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 & Zhou (2025) present a two-phase agentic system for Android vulnerability discovery and validation. Lee et al. (2025b); Zhu et al. (2025c); 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 Generation & Adaptation

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 & 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, a multi-agent framework that automatically rebuilds environments and generates verifiable exploits, successfully reproducing 428 of 841 CVEs across 22 programming languages and 141 CWE categories. 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)

This subsection describes blue-team applications of LLM agents for continuous monitoring, threat detection, incident response, threat hunting, and automated patching.

2.2.1 Autonomous Threat Detection & Incident Response

This area studies agentic SOC frameworks that monitor alerts, analyze threats, and execute response playbooks. Tellache et al. (2025) propose a RAG-based agent combining CTI and SIEM data for

automated triage to generate contextually relevant mitigation strategies against Advanced Persistent Threats (APTs). Lin et al. (2025b) propose IRCopilot to enhance incident response reliability with role-based agents, while Wei et al. (2025) introduce a collaborative agent CORTEX that reduces false alerts by 10.7% compared to single-agent baselines on a fine-grained SOC workflow dataset. 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) benchmark LLM threat reasoning, exposing performance gaps between commercial and open-source models on malware analysis, while Molleti et al. (2024) survey log-analysis agents and highlight scalability and robustness challenges inherent to high-volume industrial logging environments.

2.2.2 Intelligent Threat Hunting

Mukherjee & Kantarcioglu (2025) introduce a provenance-forensics framework using RAG, CoT reasoning, and agent orchestration to refine and verify evidence to achieve 22%/29% higher precision/recall for threat detection on six DARPA Transparent Computing and OpTC datasets compared to SOTA PIDS. 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 on a test set of cloud-specific threat reports.

2.2.3 Automated Forensics & Root Cause Analysis

Guo et al. (2025a) introduce a repository-level auditing agent that uses memory and path-condition checks to reduce hallucinations. Alharthi & 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 & Garcia (2025) introduce CIAF, an ontology-driven framework for structuring cloud logs and assembling incident narratives.

2.2.4 Autonomous Vulnerability Remediation

Several agents are proposed for automated patch synthesis and vulnerability repair. Zhu et al. (2025b) benchmark LLM agents on real CVE repair using static-analysis tools.Bouzenia et al. (2025) introduce RepairAgent,an autonomous bug-repair pipeline that interleaves tool invocation and feedback, successfully fixing 164 Defects4J bugs, including 39 not repaired by prior techniques. Applied systems include Gemini-based patching workflow (Keller & Nowakowski, 2024) and an IaC agent (Toprani & Madisetti, 2025) that integrates automated remediation into CI/CD pipelines.

2.3 Domain-specific Applications

The section explains domain-specific agentic systems using LLMs for auditing, vulnerability detection, and policy-based hardening across sectors.

2.3.1 Cloud and Infrastructure Security

Agentic systems are being employed for securing cloud and infrastructure through automated scanning, hardening, and remediation. To safely test Cloud Security Posture Management (CSPM) remediations, Yang et al. (2025b) propose a two-phase agentic workflow: sandbox "exploration" followed by verified "exploitation". Ardebili & 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 & Madisetti (2025) describe an IaC-focused agent that auto-generates CI/CD-ready fixes for policy-compliant hardening.

2.3.2 Web and Application Security

David & 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 a runtime agent to enforce 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 & 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 safety risks.

3 Threats to Agentic Systems

The safety alignment (refusal training) of a base LLM does not reliably transfer to the agentic context (Kumar et al., 2025), which introduces a new set of agent-specific security challenges (Saha et al., 2025; Chiang et al., 2025). In this section we discuss the threat landscape targeting agentic systems as well as 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). Dong et al. (2025a) show that attackers can inject malicious records into an agent's memory bank by only interacting via queries and output observations, without any direct memory access.

Lee & 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. (2025b) 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. (2025b) 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 generate novel jailbreak prompts from human-written seeds. Anil et al. (2024) demonstrate that numerous in-context examples of harmful question answering can override a model's safety training. Robey et al. (2024) present a comprehensive exploration of jailbreak attacks on agentic robotic systems.

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 & Ribeiro, 2022; Guo et al., 2025b). Pham & 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 & 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 & Park, 2025).

3.1.5 PRE-EXECUTION COGNITIVE ATTACKS

Even before tool execution, the agent's internal state—its reasoning, planning, and reflection—is vulnerable to manipulation. **Epistemic attacks** corrupt an agent's intermediate thought steps. Gre-

shake et al. (2023) demonstrate that indirect prompt injection can force the agent to condition its next step on a hallucinated prior, so that the logic remains consistent but the agent itself becomes malicious. Yang et al. (2024) inject malicious behavior in intermediate reasoning steps (e.g. calling untrusted APIs) while keeping final outputs correct. **Teleological attacks** manipulate an agent's planning graphs and goal-directed structures. Badhe (2025) demonstrate that attackers can weaponize an agent's task decomposition logic to frame a malicious objective (e.g., "steal data") as a sequence of benign-looking subtasks ("read file," "write log") that the agent mistakes for a legitimate plan. In addition, **Metacognitive attacks** target an agent's self-correction ability. Zhou & Wang (2025) show that rewriting retrieved context into epistemic tones can manipulate an agent's verification depth and self-confidence to hurt its self-correction.

3.1.6 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 gray-box 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 & 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 benchmarks and environments designed to assess agentic vulnerabilities.

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. (2025b) 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

This section describes architectural, runtime, and formal-verification defenses that strengthen agentic systems against attacks.

4.1 Defense & Operations

Here we focus on secure-by-design frameworks that embed layered verification, isolation, and control-flow integrity into agent architectures.

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 & 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 & 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 & Li, 2024; Xiang et al., 2025; Chen & 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 & 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 & Kantarcioglu, 2025; Schwartz et al., 2025; Meng et al., 2025b; Wu et al., 2025b; Meng et al., 2025a) enable explainable detection and blue-team benchmarking. Cloud-native forensic systems like LLM-Powered Forensics (Alharthi & Yasaei, 2025), CIAF (Alharthi & Garcia, 2025), CyberSleuth (Fumero et al., 2025), and GALA (Tian et al., 2025) automate evidence extraction, reduce triage time, 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

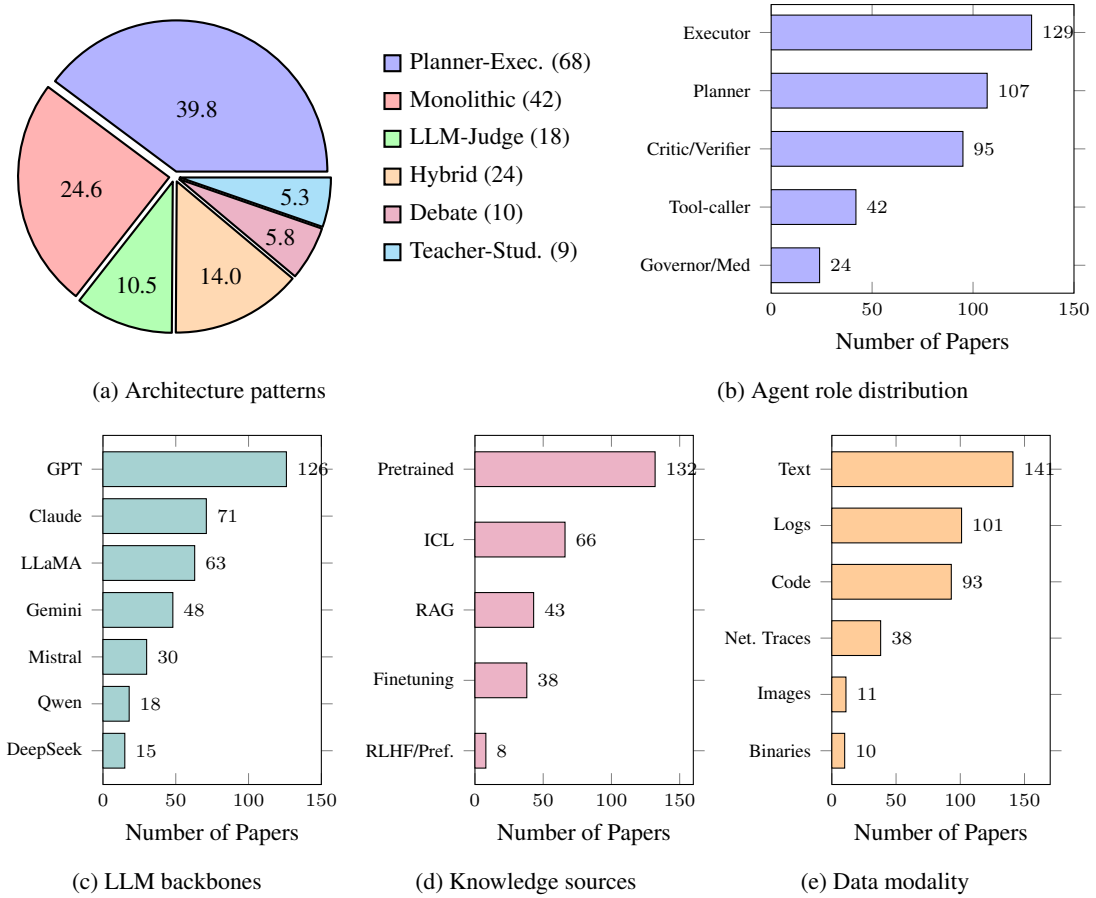
(Tur et al., 2025; Levy et al., 2025; Fu et al., 2025c) measure reliability under adversarial stress, while attack-driven frameworks like ASB, AgentHarm, and CVE-Bench (Zhang et al., 2025a; Andriushchenko et al., 2025; Zhu et al., 2025b) quantify exploitability and vulnerability reproduction. Sandboxed 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) and 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 several field-level encryption, and tamper-evident audit logging, thus 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 which reduces hallucinations and clause omissions through tool orchestration and task partitioning.



(a) Architecture patterns

(b) Agent role distribution

(c) LLM backbones

(d) Knowledge sources

(e) Data modality

Figure 2: Cross-cutting analysis of the agentic security literature under study.

5 Cross-Cutting Analysis and Trends

A cross-cutting analysis of the papers reveals clear structural patterns, as shown in Fig. 2.

Architecture, autonomy and role. The field is shifting towards planner–executor architectures (39.8%) and hybrid models (14%). It reflects a growing appreciation of decomposed cognitive pipelines, where planning, execution, and verification can be modularized to improve interpretability and debugging. More than half of the works (71 papers) implement bounded automation, eliminating the need for non-scalable human approvals. The dominance of executor and planner roles (129 papers) reflects the field's operational emphasis on task decomposition and control. Critics/verifiers appear in 95 papers. Additionally, the growing number of tool-caller (42) and governor/mediator (24) agents signifies a fundamental shift from monolithic reasoning to layered, self-checking collectives designed for explicit self-regulation and ethical alignment.

LLM backbones. GPT-family of models dominate by appearing in 83% of studies, followed by Claude, LLaMA and Gemini. Except for LLaMA, other open-weight models like Mistral, Qwen and Deepseek are in the minority, suggesting a lack of trust in their agentic capabilities. Moreover, model-specific alignment differences create fragmentation: safety fine-tuning and evaluation pipelines are rarely transferable, hindering cross-model generalization and reproducibility.

Modalities. Input modality spectrum is dominated by text, logs and codes. Although images, network traces and binaries are often tied to security vulnerabilities and intrusion, such as in browser-use agents, these non-textual modalities are underexplored. This research gap also presents a promising area for future work.

Knowledge source. Pretrained knowledge-bases dominate agentic workflows (132 papers). ICL and RAG show partial adoption, while fine-tuning, RL, and preference learning remain niche. This imbalance suggests a community preference for lightweight deployment over continual learning, which is practical for agents but potentially insecure in dynamic threat environments. It also provides future research direction in securing RAG pipelines with verified provenance, incremental fine-tuning, and model distillation.

Benchmark Fragmentation. There is a lack of consistency in evaluation methods. Some like AgentDojo rely on environment-state based evaluation, others like JAWS-BENCH use LLM-as-judge (JAWS-BENCH), and some use execution-grounded metrics. This fragmentation makes cross-benchmark comparison nearly impossible. There is also an imbalance in attack surface coverage, as prompt injection dominates benchmarks. Other attack types like memory poisoning, privilege escalation, and cognitive manipulation are underrepresented, although they are receiving more attention in recent studies. Most benchmarks target generic agent tasks (email, banking, web browsing); domain-specific benchmarks (healthcare, finance, legal) remain scarce despite high-stakes deployment in these areas. Smaller benchmarks (InjecAgent: 1,054 cases, SafeArena: 500 tasks) enable deeper attack categorization, while larger benchmarks (ASB: 400+ tools) provide breadth but often lack fine-grained threat taxonomy.

6 Conclusion

In this survey we explore the current landscape of agentic security literature. A deeper analysis shows the prevalence of multi-agent systems over monolithic architecture, the de-facto status of GPT models as the core of agentic systems, and the community preference of pre-trained knowledge for practical deployment compared to fine-tuning or RAG based approaches. Future works in this domain should focus on the challenges in cross-domain systems, design balanced unified benchmarks, and prioritize defense techniques with provable safety guaranties.

Reproducibility and Ethics. To ensure reproducibility, we have described the paper collection methodology in Appendix A, including all necessary information regarding (1) automated database search with exact search query structure; (2) manually investigated proceedings; (3) exact inclusion and exclusion criteria; and (4) snowballing efforts. In terms of ethics, we do not present any novel attacks or executable exploits; we only analyze peer-reviewed literature. While we acknowledge the risks, we believe a transparent academic study of vulnerabilities and countermeasures are essential for ensuring the safety and security of agentic systems.

REFERENCES

Ibrahim Adabara, Bashir Olaniyi Sadiq, Aliyu Nuhu Shuaibu, Yale Ibrahim Danjuma, and Venkateswarlu Maninti. Trustworthy agentic ai systems: A cross-layer review of architectures, threat models, and governance strategies for real-world deployment. *F1000Research*, 14: 905, 2025. doi: 10.12688/f1000research.144501.1. URL https://f1000research.com/articles/14-905/pdf.

Dalal Alharthi and Ivan Roberto Kawaminami Garcia. Cloud investigation automation framework (ciaf): An ai-driven approach to cloud forensics, 2025. URL https://arxiv.org/abs/2510.00452.

Dalal Alharthi and Rozhin Yasaei. Llm-powered automated cloud forensics: From log analysis to investigation. In *2025 IEEE 18th International Conference on Cloud Computing (CLOUD)*, pp. 12–22, 2025. doi: 10.1109/CLOUD67622.2025.00012.

Meysam Alizadeh, Zeynab Samei, Daria Stetsenko, and Fabrizio Gilardi. Simple prompt injection attacks can leak personal data observed by llm agents during task execution, 2025. URL https://arxiv.org/abs/2506.01055.

Amazon Web Services. Securing amazon bedrock agents: Safeguarding against indirect prompt injections, 2024. 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. Agentharm: A benchmark for measuring harmfulness of LLM agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AC5n7xHuR1.

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, Yuntao Bai, Trenton Bricken, Timothy Maxwell, Nicholas Schiefer, James Sully, Alex Tamkin, Tamera Lanham, Karina Nguyen, Tomasz Korbak, Jared Kaplan, Deep Ganguli, Samuel R. Bowman, Ethan Perez, Roger Baker Grosse, and David Duvenaud. Many-shot jailbreaking. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=cw5mgd71jW.

Mohsen Seyedkazemi Ardebili and Andrea Bartolini. Kubeintellect: A modular llm-orchestrated agent framework for end-to-end kubernetes management, 2025. URL https://arxiv.org/abs/2509.02449.

Shubhi Asthana, Bing Zhang, Ruchi Mahindru, Chad DeLuca, Anna Lisa Gentile, and Sandeep Gopisetty. Deploying privacy guardrails for llms: A comparative analysis of real-world applications, 2025. URL https://arxiv.org/abs/2501.12456.

Sanket Badhe. Scamagents: How ai agents can simulate human-level scam calls, 2025. URL https://arxiv.org/abs/2508.06457.

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. Doomarena: A framework for testing AI agents against evolving security threats. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=GanmYQ0RpE.

Alexander Bondarenko, Denis Volk, Dmitrii Volkov, and Jeffrey Ladish. Demonstrating specification gaming in reasoning models, 2025. URL https://arxiv.org/abs/2502.13295.

Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. RepairAgent: An Autonomous, LLM-Based Agent for Program Repair . In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pp. 2188–2200, Los Alamitos, CA, USA, May 2025. IEEE Computer Society. doi: 10.1109/ICSE55347.2025.00157. URL https://doi.ieeecomputersociety.org/10.1109/ICSE55347.2025.00157.

Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. Scaling trends for data poisoning in llms. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26):27206–27214, Apr. 2025. doi: 10.1609/aaai.v39i26.34929. URL https://ojs.aaai.org/index.php/AAAI/article/view/34929.

Jizhou Chen and Samuel Lee Cong. Agentguard: Repurposing agentic orchestrator for safety evaluation of tool orchestration, 2025. URL https://arxiv.org/abs/2502.09809.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. Agentpoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Y841BRW9rY.

Zheng Chen and Buhui Yao. Pseudo-conversation injection for llm goal hijacking, 2024. URL https://arxiv.org/abs/2410.23678.

Jeffrey Yang Fan Chiang, Seungjae Lee, Jia-Bin Huang, Furong Huang, and Yizheng Chen. 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*, 2025. URL https://openreview.net/forum?id=4KoMbO2RJ9.

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

Isaac David and Arthur Gervais. Multi-agent penetration testing ai for the web (mapta), 2025. URL https://arxiv.org/abs/2508.20816.

Christian Schroeder de Witt. Open challenges in multi-agent security: Towards secure systems of interacting ai agents, 2025. URL https://arxiv.org/abs/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, Shengye Wan, Spencer Whitman, Sven Krasser, and Joshua Saxe. Cybersoceval: Benchmarking llms capabilities for malware analysis and threat intelligence reasoning, 2025. URL https://arxiv.org/abs/2509.20166.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. 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*, 2024. URL https://openreview.net/forum?id=m1YYAQjO3w.

Edoardo Debenedetti, Ilia Shumailov, Tianqi Fan, Jamie Hayes, Nicholas Carlini, Daniel Fabian, Christoph Kern, Chongyang Shi, Andreas Terzis, and Florian Tramèr. Defeating prompt injections by design, 2025. URL https://arxiv.org/abs/2503.18813.

Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. PentestGPT: Evaluating and harnessing large language models for automated penetration testing. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 847–864, Philadelphia, PA, August 2024a. USENIX Association. ISBN 978-1-939133-44-1. URL https://www.usenix.org/conference/usenixsecurity24/presentation/deng.

Yinlin Deng, Chunqiu Steven Xia, Haoran Peng, Chenyuan Yang, and Lingming Zhang. 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, pp. 423–435, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702211. doi: 10.1145/3597926.3598067. URL https://doi.org/10.1145/3597926.3598067.

Yinlin Deng, Chunqiu Steven Xia, Chenyuan Yang, Shizhuo Dylan Zhang, Shujing Yang, and Lingming Zhang. 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, 2024b. Association for Computing Machinery. ISBN 9798400702174. doi: 10.1145/3597503.3623343. URL https://doi.org/10.1145/3597503.3623343.

Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. Ai agents under threat: A survey of key security challenges and future pathways, 2024c. URL https://arxiv.org/abs/2406.02630.

Fatih Deniz, Dorde Popovic, Yazan Boshmaf, Euisuh Jeong, Minhaj Ahmad, Sanjay Chawla, and Issa Khalil. aixamine: Simplified llm safety and security, 2025. URL https://arxiv.org/abs/2504.14985.

Alaeddine Diaf, Abdelaziz Amara Korba, Nour Elislem Karabadji, and Yacine Ghamri-Doudane. Bartpredict: Empowering iot security with llm-driven cyber threat prediction, 2025. URL https://arxiv.org/abs/2501.01664.

Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen Xiang. Memory injection attacks on LLM agents via query-only interaction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=QINnsnppv8.

Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen Xiang. A practical memory injection attack against llm agents, 2025b. URL https://arxiv.org/abs/2503.03704.

Mohamad Fakih, Rahul Dharmaji, Halima Bouzidi, Gustavo Quiros Araya, Oluwatosin Ogundare, and Mohammad Abdullah Al Faruque. Llm4cve: Enabling iterative automated vulnerability repair with large language models, 2025. URL https://arxiv.org/abs/2501.03446.

Richard Fang, Rohan Bindu, Akul Gupta, and Daniel Kang. Llm agents can autonomously exploit one-day vulnerabilities, 2024. URL https://arxiv.org/abs/2404.08144.

Neil Fendley, Edward W. Staley, Joshua Carney, William Redman, Marie Chau, and Nathan Drenkow. A systematic review of poisoning attacks against large language models, 2025. URL https://arxiv.org/abs/2506.06518.

Jiayi Fu, Xuandong Zhao, Chengyuan Yao, Heng Wang, Qi Han, and Yanghua Xiao. Reward shaping to mitigate reward hacking in rlhf, 2025a. URL https://arxiv.org/abs/2502.18770.

Tingchen Fu, Mrinank Sharma, Philip Torr, Shay B Cohen, David Krueger, and Fazl Barez. Poisonbench: Assessing large language model vulnerability to poisoned preference data. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=21kAulloDG.

Yuchuan Fu, Xiaohan Yuan, and Dongxia Wang. Ras-eval: A comprehensive benchmark for security evaluation of llm agents in real-world environments, 2025c. URL https://arxiv.org/abs/2506.15253.

Stefano Fumero, Kai Huang, Matteo Boffa, Danilo Giordano, Marco Mellia, Zied Ben Houidi, and Dario Rossi. Cybersleuth: Autonomous blue-team llm agent for web attack forensics, 2025. URL https://arxiv.org/abs/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. Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents, 2024. URL https://arxiv.org/abs/2411.09523.

Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. Mart: Improving llm safety with multi-round automatic red-teaming, 2023. URL https://arxiv.org/abs/2311.07689.

Luca Gioacchini, Marco Mellia, Idilio Drago, Alexander Delsanto, Giuseppe Siracusano, and Roberto Bifulco. Autopenbench: Benchmarking generative agents for penetration testing, 2024. URL https://arxiv.org/abs/2410.03225.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, AISec '23, pp. 79–90, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702600. doi: 10.1145/3605764.3623985. URL https://doi.org/10.1145/3605764.3623985.

Jinyao Guo, Chengpeng Wang, Xiangzhe Xu, Zian Su, and Xiangyu Zhang. Repoaudit: An autonomous LLM-agent for repository-level code auditing. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=TXcifVbFpG.

Qiming Guo, Jinwen Tang, and Xingran Huang. Attacking llms and ai agents: Advertisement embedding attacks against large language models, 2025b. URL https://arxiv.org/abs/2508.17674.

Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, and Zhaozhuo Xu. Llm multi-agent systems: Challenges and open problems, 2025. URL https://arxiv.org/abs/2402.03578.

Andreas Happe and Jürgen Cito. Can llms hack enterprise networks? autonomous assumed breach penetration-testing active directory networks. *ACM Trans. Softw. Eng. Methodol.*, September 2025a. ISSN 1049-331X. doi: 10.1145/3766895. URL https://doi.org/10.1145/3766895. Just Accepted.

Andreas Happe and Jürgen Cito. On the surprising efficacy of llms for penetration-testing, 2025b. URL https://arxiv.org/abs/2507.00829.

Pengfei He, Yuping Lin, Shen Dong, Han Xu, Yue Xing, and Hui Liu. Red-teaming LLM multi-agent systems via communication attacks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6726–6747, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.349. URL https://aclanthology.org/2025.findings-acl.349/.

Xu He, Di Wu, Yan Zhai, and Kun Sun. Sentinelagent: Graph-based anomaly detection in multi-agent systems, 2025b. URL https://arxiv.org/abs/2505.24201.

Yifeng He, Ethan Wang, Yuyang Rong, Zifei Cheng, and Hao Chen. Security of ai agents, 2024. URL https://arxiv.org/abs/2406.08689.

Julius Henke. Autopentest: Enhancing vulnerability management with autonomous llm agents, 2025. URL https://arxiv.org/abs/2505.10321.

Jinwei HU, Yi DONG, Zhengtao DING, and Xiaowei HUANG. Enhancing robustness of llm-driven multi-agent systems through randomized smoothing. *Chinese Journal of Aeronautics*, pp. 103779, 2025. ISSN 1000-9361. doi: https://doi.org/10.1016/j.cja.2025.103779. URL https://www.sciencedirect.com/science/article/pii/S1000936125003851.

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, Jiwei Li, Yuchen Eleanor Jiang, Wangchunshu Zhou, Guoyin Wang, Keting Yin, Zhou Zhao, Hongxia Yang, Fan Wu, Shengyu Zhang, and Fei Wu. Os agents: A survey on mllm-based agents for computer, phone and browser use, 2025. URL https://arxiv.org/abs/2508.04482.

Isamu Isozaki, Manil Shrestha, Rick Console, and Edward Kim. Towards automated penetration testing: Introducing llm benchmark, analysis, and improvements, 2024. URL https://arxiv.org/abs/2410.17141.

Feiran Jia, Tong Wu, Xin Qin, and Anna Squicciarini. The task shield: Enforcing task alignment to defend against indirect prompt injection in LLM agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29680–29697, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1435. URL https://aclanthology.org/2025.acl-long.1435/.

Yongrae Jo and Chanik Park. Byzantine-robust decentralized coordination of llm agents, 2025. URL https://arxiv.org/abs/2507.14928.

Mintong Kang and Bo Li. r^2-guard: Robust reasoning enabled llm guardrail via knowledge-enhanced logical reasoning, 2024. URL https://arxiv.org/abs/2407.05557.

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

Jun Kevin and Pujianto Yugopuspito. Smartllm: Smart contract auditing using custom generative ai, 2025. URL https://arxiv.org/abs/2502.13167.

Juhee Kim, Woohyuk Choi, and Byoungyoung Lee. Prompt flow integrity to prevent privilege escalation in llm agents, 2025. URL https://arxiv.org/abs/2503.15547.

Ronny Ko, Jiseong Jeong, Shuyuan Zheng, Chuan Xiao, Tae-Wan Kim, Makoto Onizuka, and Won-Yong Shin. Seven security challenges that must be solved in cross-domain multi-agent llm systems, 2025. URL https://arxiv.org/abs/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. A survey of llm-driven ai agent communication: Protocols, security risks, and defense countermeasures, 2025a. URL https://arxiv.org/abs/2506.19676.

He Kong, Die Hu, Jingguo Ge, Liangxiong Li, Tong Li, and Bingzhen Wu. Vulnbot: Autonomous penetration testing for a multi-agent collaborative framework, 2025b. URL https://arxiv.org/abs/2501.13411.

Panagiotis Kouvaros, Alessio Lomuscio, Edoardo Pirovano, and Hashan Punchihewa. Formal verification of open multi-agent systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, pp. 179–187, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.

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. Aligned LLMs are not aligned browser agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NsFZZU9gvk.

Christine P. Lee, David Porfirio, Xinyu Jessica Wang, Kevin Chenkai Zhao, and Bilge Mutlu. 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, 2025a. Association for Computing Machinery. ISBN 9798400713941. doi: 10.1145/3706598.3714113. URL https://doi.org/10.1145/3706598.3714113.

Donghyun Lee and Mo Tiwari. Prompt infection: Llm-to-llm prompt injection within multi-agent systems, 2024. URL https://arxiv.org/abs/2410.07283.

Hwiwon Lee, Ziqi Zhang, Hanxiao Lu, and Lingming Zhang. Sec-bench: Automated benchmarking of llm agents on real-world software security tasks, 2025b. URL https://arxiv.org/abs/2506.11791.

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

Ang Li, Yin Zhou, Vethavikashini Chithrra Raghuram, Tom Goldstein, and Micah Goldblum. Commercial llm agents are already vulnerable to simple yet dangerous attacks, 2025a. URL https://arxiv.org/abs/2502.08586.

Evan Li, Tushin Mallick, Evan Rose, William Robertson, Alina Oprea, and Cristina Nita-Rotaru. Ace: A security architecture for llm-integrated app systems, 2025b. URL https://arxiv.org/abs/2504.20984.

Simin Li, Jun Guo, Jingqiao Xiu, Ruixiao Xu, Xin Yu, Jiakai Wang, Aishan Liu, Yaodong Yang, and Xianglong Liu. Byzantine robust cooperative multi-agent reinforcement learning as a bayesian game. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=z6KS9D1dxt.

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, He Zhu, Dingfeng Shi, Piaohong Wang, Yeyi Guan, Xiangru Tang, Minghao Liu, Yuchen Eleanor Jiang, Jian Yang, Jiaheng Liu, Ge Zhang, and Wangchunshu Zhou. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl, 2025c. URL https://arxiv.org/abs/2508.13167.

Wenhao Li, Selvakumar Manickam, Yung wey Chong, and Shankar Karuppayah. Phishdebate: An llm-based multi-agent framework for phishing website detection, 2025d. URL https://arxiv.org/abs/2506.15656.

Ziyang Li, Saikat Dutta, and Mayur Naik. IRIS: LLM-assisted static analysis for detecting security vulnerabilities. In *The Thirteenth International Conference on Learning Representations*, 2025e. URL https://openreview.net/forum?id=9LdJDU7E91.

Liang Lin, Zhihao Xu, Xuehai Tang, Shi Liu, Biyu Zhou, Fuqing Zhu, Jizhong Han, and Songlin Hu. Paper summary attack: Jailbreaking llms through llm safety papers, 2025a. URL https://arxiv.org/abs/2507.13474.

Xihuan Lin, Jie Zhang, Gelei Deng, Tianzhe Liu, Xiaolong Liu, Changcai Yang, Tianwei Zhang, Qing Guo, and Riqing Chen. Ircopilot: Automated incident response with large language models, 2025b. URL https://arxiv.org/abs/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. 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, 2025a. USENIX Association. ISBN 978-1-939133-52-6.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=7Jwpw4qKkb.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications, 2024b. URL https://arxiv.org/abs/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. Secure multi-llm agentic ai and agentification for edge general intelligence by zero-trust: A survey, 2025b. URL https://arxiv.org/abs/2508.19870.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and benchmarking prompt injection attacks and defenses. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, SEC '24, USA, 2024c. USENIX Association. ISBN 978-1-939133-44-1.

Zefang Liu. Autobnb: Multi-agent incident response with large language models. In *2025 13th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–6, 2025. doi: 10.1109/ISDFS65363.2025.11012055.

Weidi Luo, Shenghong Dai, Xiaogeng Liu, Suman Banerjee, Huan Sun, Muhao Chen, and Chaowei Xiao. Agrail: A lifelong agent guardrail with effective and adaptive safety detection, 2025. URL https://arxiv.org/abs/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. xoffense: An ai-driven autonomous penetration testing framework with offensive knowledge-enhanced llms and multi agent systems, 2025. URL https://arxiv.org/abs/2509.13021.

Matteo Lupinacci, Francesco Aurelio Pironti, Francesco Blefari, Francesco Romeo, Luigi Arena, and Angelo Furfaro. The dark side of llms: Agent-based attacks for complete computer takeover, 2025. URL https://arxiv.org/abs/2507.06850.

Wei Ma, Daoyuan Wu, Yuqiang Sun, Tianwen Wang, Shangqing Liu, Jian Zhang, Yue Xue, and Yang Liu. Combining fine-tuning and llm-based agents for intuitive smart contract auditing with justifications, 2024. URL https://arxiv.org/abs/2403.16073.

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, Jingfeng Zhang, Xudong Han, Haonan Li, Jun Sun, Cong Wang, Jindong Gu, Baoyuan Wu, Siheng Chen, Tianwei Zhang, Yang Liu, Mingming Gong, Tongliang Liu, Shirui Pan, Cihang Xie, Tianyu Pang, Yinpeng Dong, Ruoxi Jia, Yang Zhang, Shiqing Ma, Xiangyu Zhang, Neil Gong, Chaowei Xiao, Sarah Erfani, Tim Baldwin, Bo Li, Masashi Sugiyama, Dacheng Tao, James Bailey, and Yu-Gang Jiang. Safety at scale: A comprehensive survey of large model and agent safety. *Foundations and Trends® in Privacy and Security*, 8(3-4):254–469, 2025. ISSN 2474-1558. doi: 10.1561/3300000051. URL http://dx.doi.org/10.1561/3300000051.

Kai Mei, Xi Zhu, Wujiang Xu, Wenyue Hua, Mingyu Jin, Zelong Li, Shuyuan Xu, Ruosong Ye, Yingqiang Ge, and Yongfeng Zhang. Aios: Llm agent operating system. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/pdf/2403.16971.

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

Yuqiao Meng, Luoxi Tang, Feiyang Yu, Jinyuan Jia, Guanhua Yan, Ping Yang, and Zhaohan Xi. Uncovering vulnerabilities of llm-assisted cyber threat intelligence, 2025a. URL https://arxiv.org/abs/2509.23573.

Yuqiao Meng, Luoxi Tang, Feiyang Yu, Xi Li, Guanhua Yan, Ping Yang, and Zhaohan Xi. Benchmarking llm-assisted blue teaming via standardized threat hunting, 2025b. URL https://arxiv.org/abs/2509.23571.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling, 2024. URL https://arxiv.org/abs/2402.09345.

Ivan Milev, Mislav Balunović, Maximilian Baader, and Martin Vechev. Toolfuzz – automated agent tool testing, 2025. URL https://arxiv.org/abs/2503.04479.

Ramasankar Molleti, Vinod Goje, Puneet Luthra, and Prathap Raghavan. Automated threat detection and response using llm agents. *World Journal of Advanced Research and Reviews*, 24:079–090, 11 2024. doi: 10.30574/wjarr.2024.24.2.3329.

Mykyta Mudryi, Markiyan Chaklosh, and Grzegorz Wójcik. The hidden dangers of browsing ai agents, 2025. URL https://arxiv.org/abs/2505.13076.

Kunal Mukherjee and Murat Kantarcioglu. Llm-driven provenance forensics for threat investigation and detection, 2025. URL https://arxiv.org/abs/2508.21323.

Lajos Muzsai, David Imolai, and András Lukács. Hacksynth: Llm agent and evaluation framework for autonomous penetration testing, 2024. URL https://arxiv.org/abs/2412.01778.

Vineeth Sai Narajala and Om Narayan. Securing agentic ai: A comprehensive threat model and mitigation framework for generative ai agents, 2025. URL https://arxiv.org/abs/2504.19956.

Subash Neupane, Shaswata Mitra, Sudip Mittal, and Shahram Rahimi. Towards a hipaa compliant agentic ai system in healthcare, 2025. URL https://arxiv.org/abs/2504.17669.

Tomas Nieponice, Veronica Valeros, and Sebastian Garcia. Aracne: An llm-based autonomous shell pentesting agent, 2025. URL https://arxiv.org/abs/2502.18528.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *Deep RL Workshop NeurIPS 2021*, 2021. URL https://openreview.net/forum?id=mp1AstNFvQ5.

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. Why do multiagent systems fail? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. URL https://openreview.net/forum?id=wM521FqPvI.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models, 2022. URL https://arxiv.org/abs/2202.03286.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models, 2022. URL https://arxiv.org/abs/2211.09527.

Viet Pham and Thai Le. Cain: Hijacking llm-humans conversations via malicious system prompts, 2025. URL https://arxiv.org/abs/2505.16888.

Rafiqul Rabin, Jesse Hostetler, Sean McGregor, Brett Weir, and Nick Judd. Sandboxeval: Towards securing test environment for untrusted code, 2025. URL https://arxiv.org/abs/2504.00018.

Melissa Kazemi Rad, Huy Nghiem, Sahil Wadhwa, Andy Luo, and Mohammad Shahed Sorower. 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)*, 2025. URL https://openreview.net/forum?id=UNPzbCKovl.

Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems, 2025. URL https://arxiv.org/abs/2506.04133.

Alexander Robey, Zachary Ravichandran, Vijay Kumar, Hamed Hassani, and George J. Pappas. Jailbreaking llm-controlled robots, 2024. URL https://arxiv.org/abs/2410.13691.

Ron F. Del Rosario, Klaudia Krawiecka, and Christian Schroeder de Witt. Architecting resilient llm agents: A guide to secure plan-then-execute implementations, 2025. URL https://arxiv.org/abs/2509.08646.

Bikash Saha and Sandeep Kumar Shukla. Malgen: A generative agent framework for modeling malicious software in cybersecurity, 2025. URL https://arxiv.org/abs/2506.07586.

Shoumik Saha, Jifan Chen, Sam Mayers, Sanjay Krishna Gouda, Zijian Wang, and Varun Kumar. Breaking the code: Security assessment of ai code agents through systematic jailbreaking attacks, 2025. URL https://arxiv.org/abs/2510.01359.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Yacmpz84TH.

Yuval Schwartz, Lavi Ben-Shimol, Dudu Mimran, Yuval Elovici, and Asaf Shabtai. Llmcloudhunter: Harnessing llms for automated extraction of detection rules from cloud-based cti. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 1922–1941, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712746. doi: 10.1145/3696410.3714798. URL https://doi.org/10.1145/3696410.3714798.

Zedian Shao, Hongbin Liu, Jaden Mu, and Neil Zhenqiang Gong. Enhancing prompt injection attacks to llms via poisoning alignment. 2024. URL https://api.semanticscholar.org/CorpusID:273502594.

Mohammed A. Shehab. Agentic-ai healthcare: Multilingual, privacy-first framework with mcp agents. *arXiv preprint*, arXiv:2510.02325, 2025. URL https://arxiv.org/abs/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. Pentestagent: Incorporating llm agents to automated penetration testing. In *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '25, pp. 375–391, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714108. doi: 10.1145/3708821.3733882. URL https://doi.org/10.1145/3708821.3733882.

Tianneng Shi, Jingxuan He, Zhun Wang, Linyu Wu, Hongwei Li, Wenbo Guo, and Dawn Song. Progent: Programmable privilege control for llm agents, 2025. URL https://arxiv.org/abs/2504.11703.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Brian Singer, Keane Lucas, Lakshmi Adiga, Meghna Jain, Lujo Bauer, and Vyas Sekar. On the feasibility of using llms to autonomously execute multi-host network attacks, 2025. URL https://arxiv.org/abs/2501.16466.

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

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Shuang Song, Yifei Zhang, and Neng Gao. Confront insider threat: Precise anomaly detection in behavior logs based on LLM fine-tuning. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 8589–8601, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.574/.

Izaiah Sun, Daniel Tan, and Andy Deng. Lisa technical report: An agentic framework for smart contract auditing, 2025. URL https://arxiv.org/abs/2509.24698.

Minxue Tang, Anna Dai, Louis DiValentin, Aolin Ding, Amin Hass, Neil Zhenqiang Gong, Yiran Chen, and Hai "Helen" Li. ModelGuard: Information-Theoretic defense against model extraction attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 5305–5322, Philadelphia, PA, August 2024. USENIX Association. ISBN 978-1-939133-44-1. URL https://www.usenix.org/conference/usenixsecurity24/presentation/tang.

Amine Tellache, Abdelaziz Amara Korba, Amdjed Mokhtari, Horea Moldovan, and Yacine Ghamri-Doudane. Advancing autonomous incident response: Leveraging llms and cyber threat intelligence, 2025. URL https://arxiv.org/abs/2508.10677.

Yifang Tian, Yaming Liu, Zichun Chong, Zihang Huang, and Hans-Arno Jacobsen. Gala: Can graph-augmented large language model agentic workflows elevate root cause analysis?, 2025. URL https://arxiv.org/abs/2508.12472.

Dheer Toprani and Vijay K. Madisetti. Llm agentic workflow for automated vulnerability detection and remediation in infrastructure-as-code. *IEEE Access*, 13:69175–69181, 2025. doi: 10.1109/ACCESS.2025.3560911.

Ada Defne Tur, Nicholas Meade, Xing Han Lù, Alejandra Zambrano, Arkil Patel, Esin DURMUS, Spandana Gella, Karolina Stanczak, and Siva Reddy. Safearena: Evaluating the safety of autonomous web agents. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=7TrOBcxSvy.

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. D-cipher: Dynamic collaborative intelligent multi-agent system with planner and heterogeneous executors for offensive security, 2025. URL https://arxiv.org/abs/2502.10931.

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

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. Agentspec: Customizable runtime enforcement for safe and reliable llm agents, 2025a. URL https://arxiv.org/abs/2503.18666.

Junjie Wang, Yuchao Huang, Chunyang Chen, Zhe Liu, Song Wang, and Qing Wang. Software testing with large language models: Survey, landscape, and vision. 50(4):911–936, April 2024. ISSN 0098-5589. doi: 10.1109/TSE.2024.3368208. URL https://doi.org/10.1109/TSE.2024.3368208.

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, Yu Wang, Hao Wu, Minghe Wang, Fan Zhang, Junfeng Fang, Wenjie Qu, Yue Liu, Chengwei Liu, Yifan Zhang, Qiankun Li, Chongye Guo, Yalan Qin, Zhaoxin Fan, Kai Wang, Yi Ding, Donghai Hong, Jiaming Ji, Yingxin Lai, Zitong Yu, Xinfeng Li, Yifan Jiang, Yanhui Li, Xinyu Deng, Junlin Wu, Dongxia Wang, Yihao Huang, Yufei Guo, Jen tse Huang, Qiufeng Wang, Xiaolong Jin, Wenxuan Wang, Dongrui Liu, Yanwei Yue, Wenke Huang, Guancheng Wan, Heng Chang, Tianlin Li, Yi Yu, Chenghao Li, Jiawei Li, Lei Bai, Jie Zhang, Qing Guo, Jingyi Wang, Tianlong Chen, Joey Tianyi Zhou, Xiaojun Jia, Weisong Sun, Cong Wu, Jing Chen, Xuming Hu, Yiming Li, Xiao Wang, Ningyu Zhang, Luu Anh Tuan, Guowen Xu, Jiaheng Zhang, Tianwei Zhang, Xingjun Ma, Jindong Gu, Liang Pang, Xiang Wang, Bo An, Jun Sun, Mohit Bansal, Shirui Pan, Lingjuan Lyu, Yuval Elovici, Bhavya Kailkhura, Yaodong Yang, Hongwei Li, Wenyuan Xu, Yizhou Sun, Wei Wang, Qing Li, Ke Tang, Yu-Gang Jiang, Felix Juefei-Xu, Hui Xiong, Xiaofeng Wang, Dacheng Tao, Philip S. Yu, Qingsong Wen, and Yang Liu. A comprehensive survey in llm(-agent) full stack safety: Data, training and deployment, 2025b. URL https://arxiv.org/abs/2504.15585.

Peiran Wang, Xiaogeng Liu, and Chaowei Xiao. CVE-bench: Benchmarking LLM-based software engineering agent's ability to repair real-world CVE vulnerabilities. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *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)*, pp. 4207–4224, Albuquerque, New Mexico, April 2025c. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.212. URL https://aclanthology.org/2025.naacl-long.212/.

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

Zhilong Wang, Neha Nagaraja, Lan Zhang, Hayretdin Bahsi, Pawan Patil, and Peng Liu. To protect the llm agent against the prompt injection attack with polymorphic prompt, 2025e. URL https://arxiv.org/abs/2506.05739.

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

Ziyue Wang and Liyi Zhou. Agentic discovery and validation of android app vulnerabilities, 2025. URL https://arxiv.org/abs/2508.21579.

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

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 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, 2023. Curran Associates Inc.

Bowen Wei, Yuan Shen Tay, Howard Liu, Jinhao Pan, Kun Luo, Ziwei Zhu, and Chris Jordan. Cortex: Collaborative llm agents for high-stakes alert triage, 2025. URL https://arxiv.org/abs/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. Psg-agent: Personality-aware safety guardrail for llm-based agents, 2025a. URL https://arxiv.org/abs/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. Excytin-bench: Evaluating llm agents on cyber threat investigation, 2025b. URL https://arxiv.org/abs/2507.14201.

Shihao Xia, Shuai Shao, Mengting He, Tingting Yu, Linhai Song, and Yiying Zhang. Auditgpt: Auditing smart contracts with chatgpt, 2024. URL https://arxiv.org/abs/2404.04306.

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. Guardagent: Safeguard llm agents by a guard agent via knowledge-enabled reasoning, 2025. URL https://arxiv.org/abs/2406.09187.

Wenpeng Xing, Minghao Li, Mohan Li, and Meng Han. Towards robust and secure embodied ai: A survey on vulnerabilities and attacks, 2025. URL https://arxiv.org/abs/2502.13175.

Hanxiang Xu, Shenao Wang, Ningke Li, Kailong Wang, Yanjie Zhao, Kai Chen, Ting Yu, Yang Liu, and Haoyu Wang. Large language models for cyber security: A systematic literature review. *ACM Trans. Softw. Eng. Methodol.*, September 2025a. ISSN 1049-331X. doi: 10.1145/3769676. URL https://doi.org/10.1145/3769676.

Kevin Xu, Yeganeh Kordi, Tanay Nayak, Adi Asija, Yizhong Wang, Kate Sanders, Adam Byerly, Jingyu Zhang, Benjamin Van Durme, and Daniel Khashabi. TurkingBench: A challenge benchmark for web agents. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *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)*, pp. 3694–3710, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.188. URL https://aclanthology.org/2025.naacl-long.188/.

Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. A comprehensive study of jailbreak attack versus defense for large language models, 2024. URL https://arxiv.org/abs/2402.13457.

Chenyuan Yang, Zijie Zhao, Zichen Xie, Haoyu Li, and Lingming Zhang. 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, 2025a. Association for Computing Machinery. doi: 10.1145/3731569.3764827. URL https://doi.org/10.1145/3731569.3764827.

Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. Watch out for your agents! investigating backdoor threats to LLM-based agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Nf4MHF1pi5.

Zhenning Yang, Archit Bhatnagar, Yiming Qiu, Tongyuan Miao, Patrick Tser Jern Kon, Yunming Xiao, Yibo Huang, Martin Casado, and Ang Chen. Cloud infrastructure management in the age of ai agents. *ACM SIGOPS Operating Systems Review*, 59(1), 2025b. doi: 10.1145/3759441.3759443. URL https://arxiv.org/pdf/2506.12270v1.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Ziyang Ye, Triet Huynh Minh Le, and M. Ali Babar. Llmsecconfig: An llm-based approach for fixing software container misconfigurations, 2025. URL https://arxiv.org/abs/2502.02009.

Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. 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, pp. 1809–1820, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712456. doi: 10.1145/3690624.3709179. URL https://doi.org/10.1145/3690624.3709179.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. LLM-Fuzzer: Scaling assessment of large language model jailbreaks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 4657–4674, Philadelphia, PA, August 2024. USENIX Association. ISBN 978-1-939133-44-1. URL https://www.usenix.org/conference/usenixsecurity24/presentation/yu-jiahao.

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. 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, pp. 6216–6226, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3736561. URL https://doi.org/10.1145/3711896.3736561.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.624. URL https://aclanthology.org/2024.findings-acl.624/.

Qiusi Zhan, Richard Fang, Henil Shalin Panchal, and Daniel Kang. Adaptive attacks break defenses against indirect prompt injection attacks on LLM agents. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7101–7117, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.395. URL https://aclanthology.org/2025.findings-naacl.395/.

Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents. In *ICLR*, 2025a. URL https://openreview.net/forum?id=V4y0CpX4hK.

Yanzhe Zhang and Diyi Yang. Searching for privacy risks in llm agents via simulation, 2025. URL https://arxiv.org/abs/2508.10880.

Yuyang Zhang, Kangjie Chen, Jiaxin Gao, Ronghao Cui, Run Wang, Lina Wang, and Tianwei Zhang. Towards action hijacking of large language model-based agent, 2025b. URL https://arxiv.org/abs/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. Webarena: A realistic web environment for building autonomous agents, 2024. URL https://arxiv.org/abs/2307.13854.

Xingfu Zhou and Pengfei Wang. Reasoning-style poisoning of llm agents via stealthy style transfer: Process-level attacks and runtime monitoring in rsv space, 2025. URL https://arxiv.org/abs/2512.14448.

Jie Zhu, Chihao Shen, Ziyang Li, Jiahao Yu, Yizheng Chen, and Kexin Pei. Locus: Agentic predicate synthesis for directed fuzzing, 2025a. URL https://arxiv.org/abs/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. CVE-bench: A benchmark for AI agents' ability to exploit real-world web application vulnerabilities. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=3pk0p4NGmQ.

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. Cve-bench: A benchmark for ai agents' ability to exploit real-world web application vulnerabilities, 2025c. URL https://arxiv.org/abs/2503.17332.

Yuxuan Zhu, Antony Kellermann, Akul Gupta, Philip Li, Richard Fang, Rohan Bindu, and Daniel Kang. Teams of llm agents can exploit zero-day vulnerabilities, 2025d. URL https://arxiv.org/abs/2406.01637.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/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, ACM CCS, Oakland) and AI (e.g., ACL, EMNLP, NeurIPS, ICLR, ICML) conferences from the same period.

Inclusion and Exclusion Criteria

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

Although recent works have explored various vulnerabilities in LLM agents, their scope is limited to specific aspects of agentic security or class of threats. Deng et al. (2024c) provide a good overview of prompt injection and data poisoning threats. Li et al. (2025a) demonstrate manipulations of commercial agents. Ma et al. (2025) review safety across several model families, and present many threats, defenses, and benchmarks. We take these findings a step further by treating agentic security as a layered system, covering not only threats but also defenses and downstream security applications. We also situate the threats in a broader taxonomy that explains where (pre-execution, during execution) and how (injection, manipulation, hacking) such failures occur, and explain how defenses are designed. Wang et al. (2025b) describe safety risks during model development pipeline, whereas we focus on what happens after deployment—how agents behave, interact, and defend themselves in the real world.

On the governance front, Yu et al. (2025) explore trustworthiness, including safety, privacy, fairness, and robustness, while Raza et al. (2025) discuss TRiSM (Trust, Risk, and Security Management) compliance. Our survey complements these high-level perspectives by examining the technical architectures and behavioral mechanisms required to enforce such security principles during operation.

Finally, several studies target specific technical or theoretical domains. He et al. (2024) and Kong et al. (2025a) analyze concrete defenses (e.g., sandboxing) and communication protocols, respectively, while de Witt (2025) establish a theoretical basis for multi-agent risks like collusion. While foundational, these works remain isolated within specific subsystems. We build on them by connecting these isolated defenses into a broader ecosystem, linking communication and system-level protections to higher-level coordination and control strategies.

Table 2: Comparison of Offensive and Defensive LLM Based Cybersecurity Agents

Feature	Offensive Agents (Red Teaming)	Defensive Agents (Blue Teaming and SOC)
Memory Designs	Context retention focused. Due to long attack chains, offensive agents prioritize architectures that minimize context loss, such as the Reasoning, Generation, Parsing design in PentestGPT. Task graph coordination is commonly used to preserve state across reconnaissance and exploitation phases.	Retrieval and structure focused. Defensive agents rely heavily on retrieval augmented generation and ontology driven memory to manage large scale telemetry. Examples include CIAF's ontology based cloud log structuring and ProvSEEK's use of RAG for evidence refinement and verification.
Tool Governance and Autonomy	High autonomy. Current trends favor fully autonomous execution for penetration testing, fuzzing, and exploit generation, as seen in systems such as AutoPentest and PentestGPT. Agents operate independently within sandboxed environments.	Human in the loop and assistive. In operational SOCs, LLMs primarily function as analyst copilots. Systems such as IRCopilot and CORTEX emphasize collaborative workflows, alert triage, and approval gated decision making.
Failure Modes Analysis	Planning and context failures. Common issues include breakdowns in multi step reasoning, loss of long horizon context, and susceptibility to prompt injection and jailbreaks. Agents can be coerced into unsafe autonomous malware execution.	False positives and hallucination. Major challenges include alert fatigue driven by false positives, which CORTEX explicitly targets. Additional failures include contradictions in CTI pipelines and hallucinated findings in auditing agents such as RepoAudit.
Evaluation Gaps and Trends	Evaluations largely rely on synthetic benchmarks such as HackSynth and AutoPenBench. The literature shows heavy dependence on GPT based backbones, approximately 83 percent of studies, raising concerns around reproducibility and generalization.	Benchmarks such as CyberSOCEval reveal gaps in threat reasoning. Log analysis agents face scalability and robustness challenges. Use of RAG and fine tuning remains limited relative to reliance on pretrained knowledge.

C Red-Teaming vs Blue Teaming (SOC Agents) analysis

Table 2 provide the comparative analysis between red-team and blue-teams, including memory design trends, tool governance and autonomy, failure mode analysis, and evaluation gaps.

D Benchmark Inventory

Table 3 provides a detailed analysis of the existing adversarial benchmarks.

Table 3: 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 $\sim$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 $\approx$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.