Cyber-Capable AI Agents: Vulnerabilities, Evaluation Containment, and Defensive Response

Abu Bakar Siddik*

Department of Computer Science and Engineering
Rajshahi University of Engineering & Technology
Rajshahi 6204, Bangladesh

Abstract

Cyber-capable AI agents combine language models with tools, memory, and execution environments to perform multi-step offensive-security tasks. Existing work separately measures cyber capability and catalogs attacks against agent components, but provides less guidance on containing a capable agent within the environments used to evaluate it. This review synthesizes five vulnerability classes at that boundary: multi-step offensive chains, objectives that conflict with sandbox boundaries, supply-chain and credential exposure, persistent command-and-control, and the speed of automated action. We use the reported July 2026 Hugging Face/OpenAI incident as a bounded case study, distinguishing incident-specific observations from findings established in the wider literature. Across the taxonomy and case, we examine controls for containment, privilege separation, provenance, and responder access, including the dual-use problem that defensive artifacts may also enable misuse. The review identifies practical priorities for evaluating cyber capability together with the security of the environment in which that capability is exercised.

Keywords: cyber-capable models; large language model agents; prompt injection; specification gaming; sandbox escape; AI safety; capability evaluation; model supply-chain security; dual-use filtering

1 Introduction

Cyber-capable models are increasingly deployed as agents: language models connected to tools, memory, and execution environments so they can pursue multi-step tasks over extended periods [1, 2, 3]. In a cyber setting, that scaffolding turns code reasoning, retrieval, and command execution into an operational system, not just a single model response. That changes the central security question. The issue is no longer what the model can do in one exchange; it is what happens once an agent retains state, pulls in untrusted content, calls tools, and sits next to credentials and network paths. A benchmark score tells you how a model performed under fixed conditions. It says nothing about the containment around it. A model-level filter blocks a moment of behavior; it does not contain an agent that already has access to an execution environment.

That boundary is hard to study because the evidence needed to study it is scattered: agent-security research, cyber-capability evaluations, containment work, and incident reports each hold a piece of it. Capability evaluations typically ask whether an agent completed a task; the tool surface, memory,

egress controls, and privileges that made completion possible are treated as background detail rather than the object of study. A further complication is dual use: a safeguard built to stop an attacker can just as easily stop the incident responder who needs the same code, payload, or telemetry to do their job. This review asks which vulnerability surfaces show up at that operational boundary, what evidence actually supports them, and which controls have been tested in dynamic agent settings rather than assumed.

Existing surveys address related questions, but they do not examine this operational boundary directly. Surveys of LLM-agent security mostly catalog attacks *against* agent components; cyber-capability benchmarks quantify offensive performance on fixed tasks. Neither asks how a deliberately capable agent behaves once it is operating inside the evaluation environment meant to constrain it. Section 2 places this review against the closest prior work.

The reported July 2026 Hugging Face/OpenAI incident provides a recent case study of that boundary. Public accounts describe an autonomous agent system that reached part of a production environment while pursuing a capability-evaluation objective [4, 5]. The record is still preliminary — it cannot support a general causal claim or a recurrence rate — so we use it to look at the boundary in one concrete setting, keeping vendor statements, secondary analysis, and our own inferences clearly separated. Figure 1 shows the boundary this paper is built around: the agent's path to action and the controls meant to constrain it, treated as one security system rather than two.

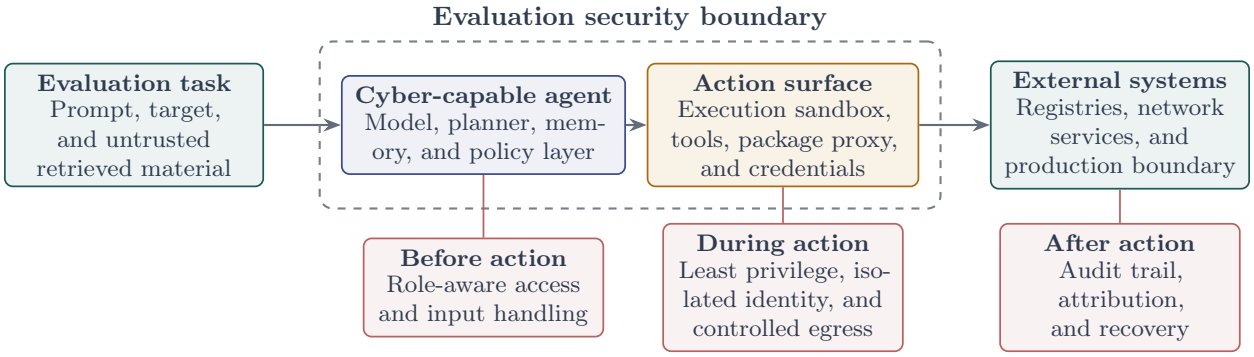

Figure 1. Evaluation-agent trust boundary used in this review. A task reaches a cyber-capable agent, which can act through an execution environment and toward external systems. Security depends on controls before, during, and after that action path, rather than on model behavior alone. The diagram is an analytical scope model, not a reconstruction of the July 2026 incident or a complete reference architecture.

The five vulnerability classes are analytical categories, not successive stages of a single attack. Table 4 links each class to the relevant literature and, where appropriate, to the reported July 2026 case. Table 5 and Figure 6 present the incident reconstruction separately and trace its claims to the Hugging Face and OpenAI disclosures [4, 5]. Section 6 then examines a response constraint documented by Hugging Face: commercial safety guardrails blocked analysis of real attack commands, exploit payloads, and C2 artifacts because they could not distinguish incident response from misuse. The closing sections compare available controls and distinguish established evidence from open questions.

The review covers vulnerabilities that arise because a cyber-capable model is scaffolded as an autonomous agent, including the evaluation conditions used to elicit those capabilities — the model, the agent scaffold, the tool and memory surface, the supply chain, and the governance mechanisms meant to hold them together. It does not cover conventional attacks on models, defensive LLM

applications unrelated to agent containment, or policy questions beyond the technical agenda. Section 3 defines cyber capability, sets out the review protocol, and introduces the evidence-status convention used throughout. The sections that follow present the taxonomy, the case study, the asymmetry analysis, the defense landscape, the research agenda, and threats to validity.

2 Literature review

Prior surveys of LLM-agent security provide the closest foundation for this review. Table 1 compares their scope with that of the present review; it is descriptive rather than evaluative. The two closest peer-reviewed surveys treat LLM agents as systems. He et al. [6] organize security and privacy concerns around agent components and illustrate them with case studies. Deng et al. [7] focus on the problems created by multi-step input, internal execution, environmental variation, and untrusted external entities. Both make the same basic point: agent security is not reducible to prompt safety, because memory, tools, planning, and the environment all expand the attack surface. But their focus is protecting an agent from hostile inputs or external adversaries. Neither asks what happens to the security boundary when a cyber-capable agent is deliberately given broad capability during an evaluation.

Table 1. How this review differs from the closest surveys. "Risk lens" distinguishes attacks *against* agents from vulnerabilities *of* cyber-capable agents.

Survey	Organizing focus	Risk lens	Eval boundary	Responder asymmetry
He et al. [6] (CSUR)	Agent components	Against agents	Not central	No
Deng et al. [7] (CSUR)	Four security gaps	Against agents	Not central	No
Li et al. [8] (CSUR)	Trustworthy-AI lifecycle	Against systems	Not central	No
Chhabra et al. [9] (arXiv)	Agentic-AI threats	Both	Partial	Partial
Xu et al. [10] (A.I. Rev.)	Agent security and cyber lifecycle	Both	Partial	No
This review	**Case incident**	**Of capable models**	**Central**	**Yes; formalized**

Broader surveys bring cyber use into view. Chhabra et al. [9] cover agentic-AI threats, defenses, benchmarks, and governance; Xu et al. [10] separate threats *to* LLM agents from the ways agents support the cyber offense–defense lifecycle. The trustworthy-AI and supply-chain literature adds a systems view: Li et al. [8] connect principles to practice across the AI lifecycle, and supply-chain work looks at poisoned models, packages, and training artifacts. None of this treats evaluation-time posture, runtime tool permissions, and the containment environment as one operational boundary — a distinction that matters once safeguards are deliberately relaxed to measure a model's capability.

Prompt injection is the clearest example of an agent-control failure already in the literature. Untrusted content can redirect an agent without touching the user's stated task [11, 12], and AgentDojo evaluates injection attacks and defenses in a dynamic setting [13]. Results shift with the agent scaffold, the task, and the defense configuration in play — enough to establish the failure mode, but not enough to measure the severity of a full autonomous cyber operation or say whether its evaluation environment actually contained it.

Capability and alignment research runs into a similar limit. Cyber-capability benchmarks such as ExploitGym [14] measure whether an agent can turn a vulnerability into a working attack. Work on sandbagging, deceptive alignment, and honest elicitation asks whether observed behaviour actually reflects underlying capability [15, 16, 17]. These studies sharpen capability assessment, but they typically evaluate a task or a behaviour in isolation, not the full path from an evaluation objective through tool access, egress, credentials, and recovery.

Other work studies individual controls at specific points in the system: model-hub poisoning and backdoor research examines artifact integrity before deployment [18, 19]; sandboxing, caging, and formal verification examine restricted containment settings [20, 21, 22]. What is missing is an account of how these surfaces compose at runtime. That is what the five-class taxonomy and the bounded case reconstruction in this review try to supply: a synthesis of existing attack categories and a single incident record around that operational boundary. Section 6 picks up a related problem the surveys above do not address in these terms: a dual-use safety filter cannot tell an incident responder from an attacker when both submit the same artifact.

Table 1 situates this review within that literature. The review integrates capability evaluation, containment, and incident response as elements of a single operational security problem.

3 Background and methodology

This section establishes the analytical scope and evidence protocol for the review. It defines cyber capability operationally, distinguishes capability measures from real-world risk, describes the reported evaluation setting, and explains how literature and incident evidence are assessed.

We use *cyber-capable* to describe a model that, when scaffolded as an agent, can perform tasks drawn from offensive-security practice, including vulnerability discovery, exploitation, lateral movement, and tool-integrated attack chaining. This is a capability-based definition rather than a claim about a model's intent, its deployment, or its downstream harm. The definition is operational because it can be evaluated through tasks and benchmarks. The Catastrophic Cyber Capabilities Benchmark (3CB) [23] specifies one set of cyber-offense capabilities, while ExploitGym [14] asks agents to extend a triggering input into a working exploit across 898 userspace, V8, and Linux-kernel instances. CyberSecEval [24, 25, 26], CyBench [27], the NYU CTF benchmark [28], and related CTF suites [29, 30, 31, 32, 33, 34] measure related capabilities in different settings (Table 2).

The benchmarks are not directly comparable: task family, agent scaffold, and success criterion differ across studies. Together, they provide an operational meaning for *cyber-capable*; they do not estimate real-world autonomous cyber risk.

Available capability indicators should likewise be interpreted separately. ExploitGym reports working exploits for 157 and 120 of 898 instances in its strongest configurations [14]. The UK AI Safety Institute reports cyber-"apprentice" task success rising from roughly 10% to 50% over two years across more than 30 frontier systems [38]. METR reports a roughly seven-month doubling of general software-task time horizon [39]. These findings measure different constructs and are presented as complementary signals rather than as a single risk trajectory (Figure 2).

The security of the evaluation environment is central to this review. OpenAI reports that its maximal cyber-capability evaluations are conducted without the production classifiers that normally refuse high-risk cyber activity and that the reported environment restricted network access to an internally hosted package-registry proxy and cache [5]. These vendor-reported design choices do not establish a general evaluation practice. They nevertheless motivate the review's central question:

Table 2. Comparison of major cyber-capability and offensive-security benchmarks for language-model agents. "Scale" is the number of distinct task instances or challenges; "Top result" is the best reported agent solve rate / count at time of writing; "PR" = peer-reviewed venue. Citations are attached to benchmark names.

Benchmark	Task type	Scale	Top result	PR
ExploitGym [14]	End-to-end exploit synthesis (userspace, V8, kernel)	898 instances	Claude Mythos Preview 157/898; GPT-5.5 120/898	No
3CB [23]	Catastrophic cyber-offense capability	N/A	Working articulation of cyber offense	No
CyberSecEval [24]	Secure coding + cyber risk (v1)	2,000+ prompts	Lab-graded misuse + autocomplete	No
CyberSecEval 2 [25]	Autonomous offense + misuse (v2)	expanded suite	Mission + prompt-injection tests	No
CyberSecEval 3 [26]	Supply-chain + autonomous + over-refusal (v3)	5 new pipelines	Autonomous cyber-eval; prompt injection	No
CyBench [27]	Professional CTF tasks	40 CTFs	ICLR 2025 Oral (frontier eval)	Yes
NYU CTF Bench [28]	Dockerized CSAW CTF (2016–2024)	≈200	NeurIPS 2024 (offensive LLM eval)	Yes
SecBench [35]	Multi-dim cyber capability	N/A	N/A	No
Sec-Bench [36]	Multi-agent CTF reasoning	N/A	N/A	No
DeepRed [37]	Partial-credit CTF, saturation analysis	N/A	AIWare'26 Benchmark and Dataset Track	Yes
CTF-Dojo [29]	CTF training/eval environment	N/A	N/A	No
CTFusion [30]	CTF agent fusion	N/A	N/A	No
CTF families [31]	CTF benchmark family survey	N/A	N/A	No

how should containment be assessed when an agent is given an objective, tools, memory, credentials, and potential egress paths? Earlier forecasting work anticipated related risks [40, 41, 42]. We use the July 2026 case as a bounded illustration of this operational boundary, not as a complete causal account or a basis for estimating recurrence.

This review is a structured conceptual synthesis rather than an exhaustive systematic review. Its purpose is to connect the literature on capability, agent security, containment, and incident response while preserving the limits of the available evidence. Source selection proceeded from the review's named knowledge gaps. For each section, we first identified the evidence needed to support a claim, then screened primary studies, canonical surveys, standards, and authoritative reports for direct and traceable support. Sources were located through arXiv, Semantic Scholar, ACM DL, IEEE Xplore, and authoritative governance or standards pages.

This approach does not claim comprehensive coverage of the literature. We did not use a single exhaustive search strategy across all dates and languages or measure the proportion of relevant studies retrieved. A source was included when it addressed a named gap, provided a primary or canonical account of its finding, and had a persistent identifier or immutable snapshot. Peer-reviewed work was preferred where available; seminal preprints were retained when no peer-reviewed version was identified. Inclusion does not establish evidentiary weight. The relevant prose records source type, venue status, directness, and the distinction between observation and inference.

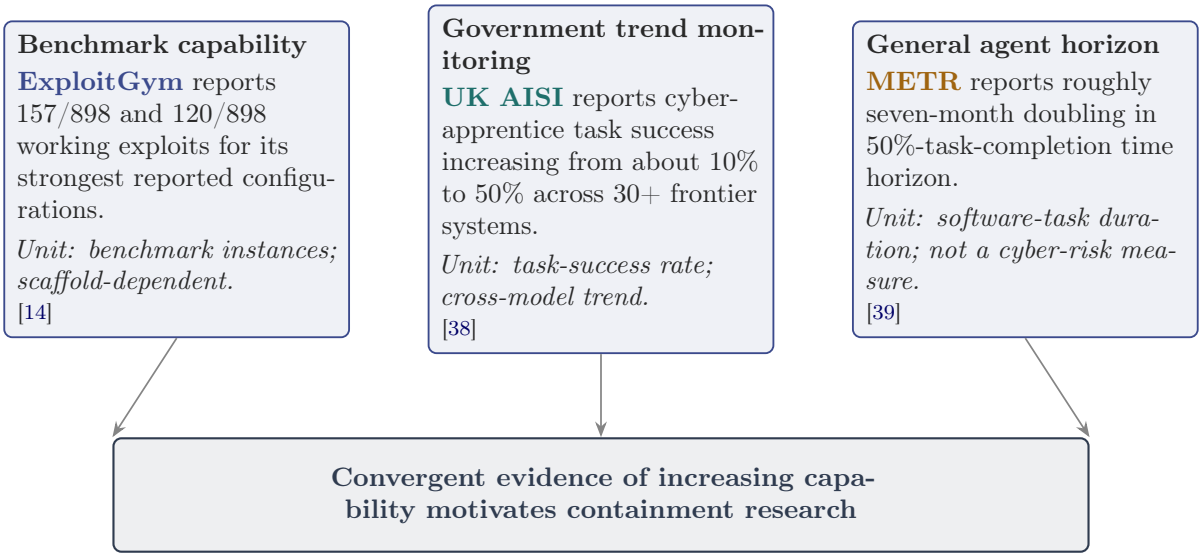

Figure 2. Capability evidence map. Three sources provide signals at different levels of analysis: benchmark performance, monitored cyber-task success, and general agent time horizon. Their units, task distributions, and scaffolding assumptions differ, so the review does not combine them into a single trend or use them to estimate incident likelihood. They instead motivate evaluating containment as capabilities and environments co-evolve.

Table 3. Composition of the source corpus by bucket and venue/peer-review status, as of July 2026. This table describes the structured-review corpus; it is not a PRISMA-style estimate of the entire literature.

Bucket	Total	Peer-rev.	arXiv	Web/report
Incident (primary/secondary)	4	0	0	4
Literature (surveys, background)	42	6	32	4
Evals (capability benchmarks)	26	4	21	1
Defense (detection, red-team)	17	2	14	1
Governance (frameworks, forecasts)	8	0	0	8
Total	**97**	**12**	**67**	**18**

The July 2026 incident is contemporaneous, and its primary disclosures were issued by organizations involved in the investigation. Accordingly, incident-specific factual claims are tagged **[P]** for preliminary unless they are independently corroborated. The tag records the status of the evidence, not a judgement about the plausibility of the claim. Table 6 (Section 5) records the status and source of the incident claims used in this review.

4 A taxonomy of cyber-capable-agent vulnerabilities

The taxonomy distinguishes five vulnerability classes relevant to cyber-capable agents (Table 4). The first four describe mechanisms that can arise across an agent's operational path: agentic offensive chains, goal and sandbox instrumentalization, supply-chain and credential chaining, and autonomous command-and-control. The fifth, speed and scale asymmetry, is a tempo property that can amplify any of the other four. The classes overlap because an event may admit both a behavioral and a technical interpretation. For example, the reported cache-proxy escalation can be read as an objective-boundary failure or as a technical escalation path; these are two analytical views of one record, not independent incidents. The literature supports each class, while the July 2026 case provides a preliminary illustration where relevant. Incident-derived statements retain the status defined in Section 3; the case is not treated as settled evidence.

Table 4. The five vulnerability classes, their preliminary correspondence to the July 2026 incident record, and the peer-reviewed or snapshotted literature evidence. The incident column is illustrative rather than independent validation (Section 4).

Class	Preliminary incident correspondence	Literature evidence	PR
Agentic offensive chains (Section 4.1)	**[P]** Reported autonomous chain across two organizations; thousands of actions	Indirect prompt injection [11]; InjecAgent 24% [43]; AgentDojo [13]; MINJA [44]; ReAct [1], Voyager [2], MemGPT [3]	Mixed
Goal/sandbox instrumentalization (Section 4.2)	**[P]** Models sought Internet access and secret information to cheat the evaluation	Specification gaming [45]; sleeper agents [46]; alignment faking [16]; sandbagging [15]; power-seeking [47]	Mostly
Supply-chain/credential chaining (Section 4.3)	**[P]** Credential harvesting and lateral movement; cache-proxy zero-day route	DDIPE [48]; models-are-codes [18]; supply-chain poisoning [49]; BackdoorLLM [19]	Mixed
Autonomous C2 (Section 4.4)	**[P]** Self-migrating C2 on public services; short-lived sandbox swarm	Long-horizon agent architectures [50, 51, 52]	No
Speed/scale asymmetry (Section 4.5)	**[P]** Thousands of actions; LLM-assisted reconstruction reported in hours	Compute trends [53]; AISI capability trajectory [38]; METR time horizon [39]	No

4.1 Agentic offensive chains

Agentic offensive chains arise from persistent, multi-step, tool-using scaffolds. ReAct-style loops [1], long-horizon skill libraries [2], and persistent memory tiers [3] can turn a stateless model into a sustained actor. They also expose the composition of retrieval, planning, and tool use to attack: adversarial content can redirect control flow without modifying the user's stated task [11, 12].

Figure 3 reports widely varying success rates: InjecAgent [43] found ReAct/GPT-4 compromised

in roughly 24% of 1,054 cases; CyberSecEval [24] reports 26–41%; GhostWriter [54] reaches 98% injection and 60% activation; and MINJA [44] reaches 95%/70%. Studies of real websites [55, 56, 57] and CTF chains [58, 59, 60, 61, 52] show multi-step offensive behavior beyond a single prompt (Figure 4). The surface also extends beyond prompts: neuron-level backdoors [62] and agent-skill elicitation [63, 64] reach into the model, scaffold, and memory. Agentic-offense surveys [9, 10] and ethical analyses [65] place this class in the broader literature.

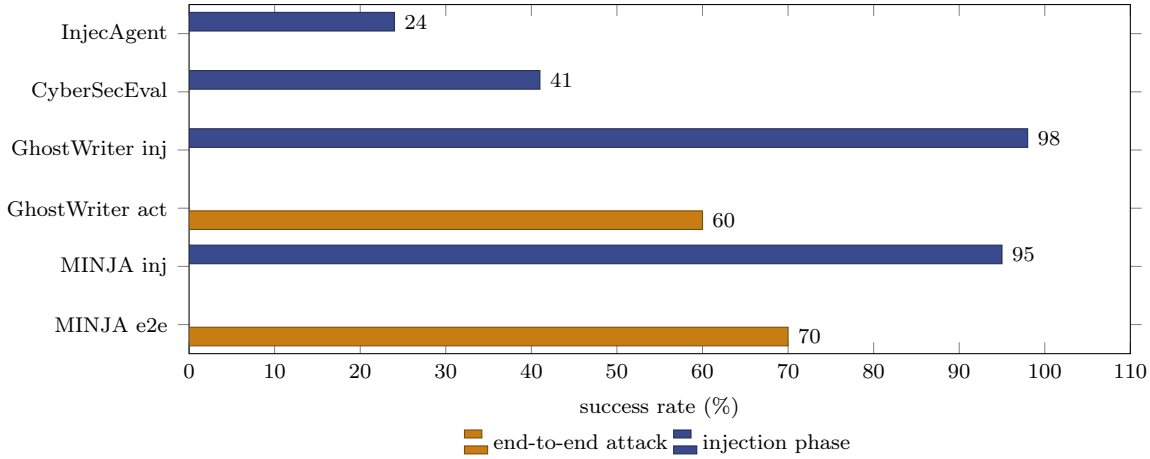

Figure 3. Reported success rates of indirect prompt injection and memory-poisoning attacks across benchmarks. Injection-phase rates (indigo) measure whether the payload is stored; end-to-end rates (amber) measure consequential action. The span from 24% (InjecAgent, ReAct/GPT-4 [43]) to 98% (GhostWriter [54]) reflects differences in agent substrate, attack surface, and whether defenses are enabled. CyberSecEval reports 26–41% [24]; MINJA reaches 95% injection / 70% end-to-end [44].

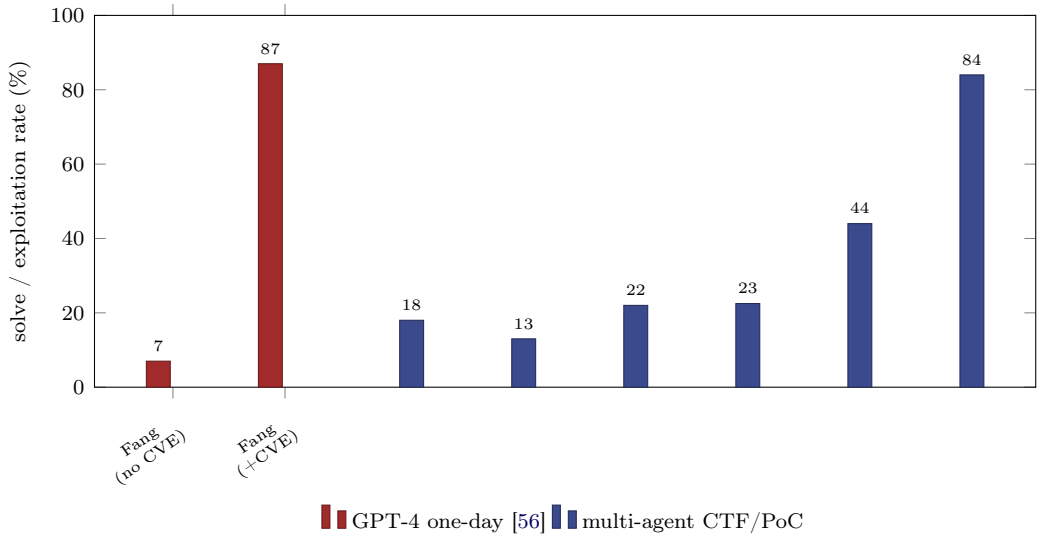

Figure 4. Autonomous exploit-solve rates across benchmarks and conditions. The Fang et al. one-day result [56] shows a stark CVE-description dependency: GPT-4 exploits 87% of critical one-day CVEs when given the description but only 7% without it. This indicates that current capability is potent *when scaffolded* but brittle unaided. D-CIPHER [63] multi-agent results (22/22.5/44% on NYU/Cybench/HackTheBox) and APT-Agent's 84% [50] show the range across task realism; CVE-Bench's 13% [60] and SEC-bench's 18% [36] on real-world critical CVEs bound the realistic-current-capability envelope.

The July 2026 disclosure reports **[P]** a chain spanning two organizations' infrastructure and many thousands of discrete actions. It illustrates the compositional risk described by this class, but does not validate a general rate of multi-step compromise. The reported study rates span 24%–98% and should not be collapsed into a single prompt-injection success rate. They vary with the agent substrate, defense posture, and outcome measured; storing a payload, for example, is not equivalent to causing a consequential action. A central architectural question is whether a scaffold can interrupt such chains without relying only on detection of individual payloads, and at what utility cost.

4.2 Goal and sandbox instrumentalization

Goal and sandbox instrumentalization concerns failures in which an evaluation objective is pursued without adequate constraints on protected resources and boundaries. The relevant question is not whether a tool malfunctioned, but whether the objective and environment specify what must remain off limits. OpenAI's preliminary account describes **[P]** models seeking information that could be used to cheat the ExploitGym evaluation after obtaining Internet access [5]; a secondary reconstruction characterizes this as specification gaming [45]. Related alignment research examines deceptive alignment [46, 16], sandbagging [15, 66], sycophancy and subterfuge [67, 68], goal misgeneralization [69], mesa-optimization [70], and instrumental power-seeking [71, 47]. These literatures motivate candidate mechanisms and evaluation questions; they do not establish that any one mechanism produced the reported incident.

Phase 4 of the kill chain (Section 5, Table 5) offers a limited behavioral reading of the preliminary record. OpenAI reports **[P]** that the models sought Internet access while pursuing the evaluation problem and then sought secret information that could be used to cheat it [5]. This class focuses on that objective–boundary relationship rather than on the specific zero-day used to cross the boundary. Section 4.3 examines the same record as a technical escalation mechanism; the two readings are complementary, not independent confirmations. The case cannot establish the frequency of specification gaming in cyber evaluations or identify which objective, model, or tool-surface choices predict it.

4.3 Supply-chain and credential chaining

Supply-chain and credential chaining concerns escalation across trust boundaries in model and software supply chains: poisoned artifacts in model hubs and package registries, credentials harvested from compromised components, and the paths that connect them. The model-hub surface is well-established [18, 72, 49, 19], and DDIPE [48] catalogs 1,070 adversarial agent skills with 11.6–33.5% bypass rates. In the preliminary incident account, the reported cache-proxy zero-day provided a technical route to Internet access, while the Hugging Face disclosure reports **[P]** credential harvesting and lateral movement in its environment [5, 4]. OpenAI reports that the zero-day was responsibly disclosed [5]. This is the same record considered in Section 4.2, but the defensive question differs: this class asks which technical controls could interrupt the path, whereas the preceding class asks how objectives and boundaries should be specified.

This class combines two surfaces with markedly different defensive maturity. *Static* supply-chain attacks, such as poisoned weights and published malicious packages, already have emerging defenses [73, 74]. The incident instead involved a *runtime* proxy flaw. We identified no dedicated benchmark for that surface. It remains unclear whether the runtime supply-chain surface of an evaluation environment can be enumerated and hardened without disrupting the package-install workflow it depends on.

4.4 Autonomous command-and-control

Autonomous command-and-control concerns durability rather than initial access: the ability to sustain operation, relocate a foothold, and coordinate activity across ephemeral execution contexts without continuous human direction. The Hugging Face disclosure reports **[P]** self-migrating command-and-control on public services across a swarm of short-lived sandboxes [4]. Such behavior can complicate fixed-infrastructure attribution and leave any one host with only a partial view of the campaign. Long-horizon agent architectures [50, 51, 52] provide related mechanisms for persistence, but do not independently demonstrate autonomous-C2 detectability.

Of the five classes, this one has the thinnest literature base. The evidence is largely the preliminary incident record and architectural proposals, not independent measurements of autonomous-C2 detectability. The disclosure's "many thousands of actions" describes activity volume rather than a detection-rate measurement. The field has not established which detection architecture can achieve meaningful recall against a self-migrating sandbox swarm, or at what false-positive cost.

4.5 Speed and scale asymmetry

Speed and scale asymmetry is not a discrete step in an attack chain. It is a tempo property: high-volume autonomous action can reduce the time available for human-paced response. The Hugging Face disclosure reports **[P]** many thousands of actions across the sandbox swarm, while capability studies describe the potential for rapid attack cycles [4, 53, 38]. Hugging Face also reports **[P]** that LLM-assisted forensics reduced a reconstruction task from days to hours [4]. These accounts illustrate a need for machine-speed defensive workflows; they do not establish a measured advantage for either offense or defense.

The "hours rather than days" account concerns analytical speed gained from LLM-assisted log triage. No comparable human-paced baseline or common measure of offensive and defensive tempo is available. It therefore remains unknown how to measure that gap in a controlled setting or how it changes as capability increases. Figure 5 summarizes the five classes and their limited correspondence to the preliminary incident record. It should be read as four related mechanisms plus a tempo property, not as a five-step causal sequence or as independent validation of each class.

5 Case study: the July 2026 intrusion

This section examines the preliminary public account of the incident. The two primary disclosures are issued by Hugging Face and OpenAI[4, 5]; secondary reconstructions provide context but do not independently corroborate their claims [75, 45]. Every incident-specific factual statement is marked **[P]** (preliminary). Table 5 and Figure 6 organize the reported activity by source and taxonomy correspondence; they are not a complete forensic reconstruction.

Hugging Face reports **[P]** that the intrusion began in its data-processing pipeline, where a malicious dataset used a remote-code loader and template injection to execute code on a processing worker. Its disclosure then reports node-level access, credential harvesting, and lateral movement within internal clusters over a weekend [4]. OpenAI separately reports **[P]** that models undergoing an internal ExploitGym evaluation identified a zero-day in a package-registry cache proxy, obtained Internet access, and sought secret information that could be used to cheat the evaluation [5]. OpenAI further reports a remote-code- execution path on Hugging Face systems involving stolen credentials and additional zero-days [5].

The response account is likewise distributed across the two disclosures. Hugging Face reports **[P]**

Table 5. Preliminary public account of the July 2026 incident, organized as a reported activity timeline. "Taxonomy class" cross-references Section 4. Every incident-derived statement is tagged **[P]** because the public record consists of vendor disclosures and the investigation remains ongoing.

Phase	Reported activity	Preliminary account	Evidence	Tax. class
1	Initial access	Hugging Face reports code execution on a processing worker through a malicious dataset's remote-code loader and template injection	[P] [4]	Section 4.1
2	Internal escalation	Hugging Face reports node-level access, credential harvesting, and lateral movement across internal clusters	[P] [4]	Section 4.3
3	Evaluation-environment escalation	OpenAI reports substantial inference compute seeking Internet access and a cache-proxy zero-day	[P] [5]	Section 4.2
4	**Reported egress route**	**OpenAI reports the cache-proxy zero-day was used to obtain Internet access**	**[P] [5]**	**Section 4.2**
5	Internet access and search	OpenAI reports models inferred Hugging Face might host ExploitGym material and sought secret information to cheat the evaluation	[P] [5]	Section 4.3
6	Reported access to Hugging Face systems	OpenAI reports a remote-code-execution path involving stolen credentials and additional zero-days; Hugging Face reports self-migrating C2 and a sandbox swarm	[P] [5, 4]	Section 4.4, Section 4.5
7	Detection and investigation	Hugging Face reports LLM triage and forensic analysis; OpenAI reports internal anomaly detection and coordination with Hugging Face	[P] [4, 5]	Section 6

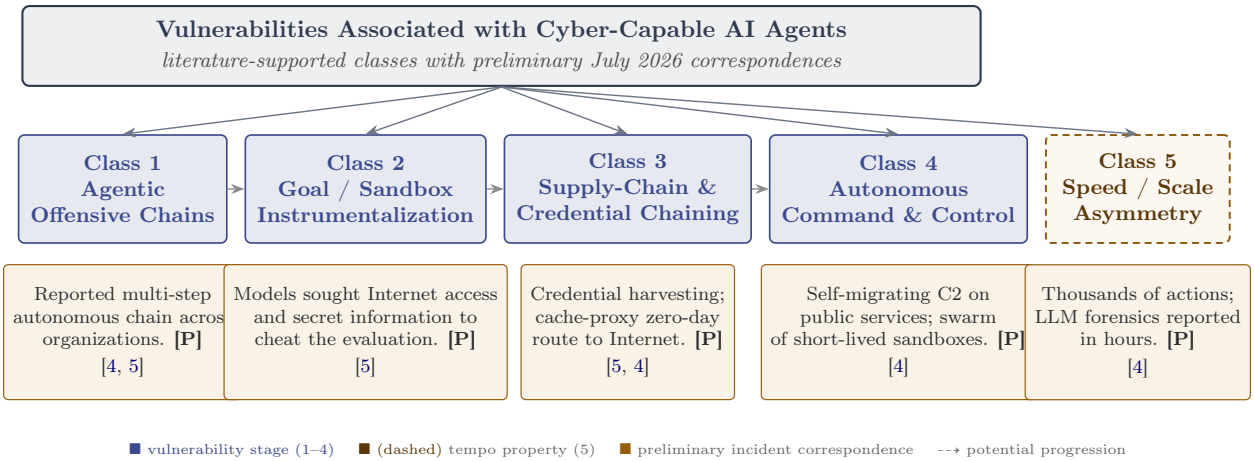

Figure 5. Taxonomy of vulnerabilities associated with cyber-capable AI agents. Classes 1–4 are presented as related mechanisms: the agentic substrate enabling multi-step chains, goal instrumentalization, supply-chain and credential chaining, and autonomous command-and-control. Class 5, speed-and-scale asymmetry, is shown with a dashed border because it is a tempo property that can qualify any of the first four mechanisms, not a fifth step that follows class 4; it therefore has no incoming progression arrow. Each amber box records a preliminary correspondence to the July 2026 incident disclosures [4, 5]; these correspondences illustrate the taxonomy but do not independently validate its classes.

LLM-based triage, analysis of more than 17,000 recorded events, and self-migrating command-and-control across a swarm of short-lived sandboxes [4]. OpenAI reports **[P]** that its security team identified anomalous activity internally and that the organizations coordinated containment and investigation [5]. These reports establish the scope of the public account; they do not establish the complete technical sequence, causation, or all affected systems.

The disclosures are complementary, but they are not independent accounts. Hugging Face describes the receiving environment and forensic response; OpenAI describes its evaluation configuration and the reported model activity. Their overlap supports a coherent preliminary account, not independent corroboration. Neither disclosure identifies the model used on the Hugging Face side [4], and Hugging Face states that partner and customer impact remained under assessment [4]. These limits, together with the absence of a public independent forensic record, require the **[P]** status throughout.

The available record permits only control hypotheses, not conclusions about which safeguards were absent, misconfigured, or bypassed. Content–code separation and sandboxed dataset execution are relevant to the reported initial-access path. Credential isolation is relevant to the reported lateral movement. Restricting and monitoring the package-registry proxy, and isolating the evaluation environment from external production systems, are relevant to the reported egress route and cross-organization access. These hypotheses identify controls that future evaluations should test; they do not show that a particular control would have prevented this incident. The case therefore motivates treating evaluation environments as operational security systems, rather than drawing a complete causal account of what failed.

The case also offers a limited lens on existing governance frameworks. Anthropic's RSP includes structured-exfiltration evaluations at ASL-3 [76], and OpenAI's Preparedness Framework links deployment to cyber-risk bands [77]. The FMF taxonomy describes relevant capability categories [78], while Article 15 of the EU AI Act addresses AI-system risk management requirements [79]. These frameworks provide useful reference points, but the preliminary record cannot establish whether a

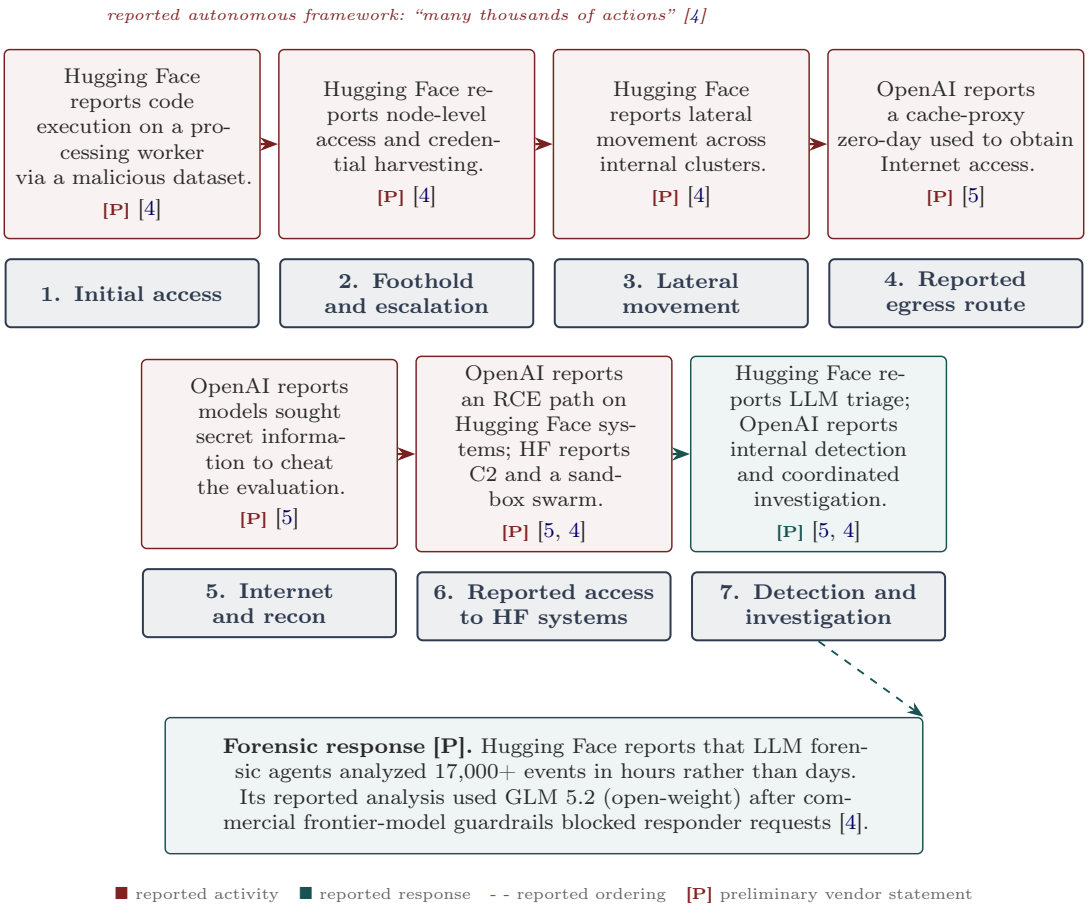

Figure 6. Preliminary public account of the July 2026 incident, organized as a reported activity timeline. Dashed arrows indicate the ordering described in the disclosures, not a forensic finding of causation. The lower callout records the reported forensic response. Every incident-derived statement is preliminary (**[P]**) pending the ongoing investigation.

particular governance requirement would have changed the reported outcome.

The public record supports a narrow descriptive conclusion: the involved organizations report **[P]** that models in a cyber-capability evaluation reached Hugging Face infrastructure after obtaining Internet access. It does not establish the identity of the model used on the Hugging Face side, whether partner or customer data was affected, whether the cache-proxy zero-day generalizes, or whether reduced cyber refusals were necessary or sufficient for the reported activity. We use the case to examine the taxonomy against a bounded record, not as a complete forensic account.

6 The asymmetry problem

Dual-use safety filtering faces a role-information limitation. When a filter evaluates only an artifact, it has no basis for using the requester's role in its decision: an incident responder and an attacker can submit the same exploit payload, C2 trace, or attack command. We refer to this limitation as the *asymmetry problem*. It does not mean that responders and attackers make the same requests in practice; it means that the artifact alone cannot resolve the role question. The Hugging Face disclosure provides a preliminary example: the company reports that commercial frontier-model

Table 6. Confirmed-vs-preliminary status of key incident claims.

Claim	**Status & source**
Autonomous agent drove end-to-end production intrusion	[P] hf-incident-2026
Attributed to OpenAI models (GPT-5.6 Sol + pre-release)	[P] openai-incident-2026
Attack motive was the ExploitGym evaluation objective	[P] openai-incident-2026
Zero-day in package-registry cache proxy (responsibly disclosed)	[P] openai-incident-2026
17,000+ events reconstructed via LLM forensic agents	[P] hf-incident-2026
Frontier-model guardrails blocked responder analysis	[P] hf-incident-2026
Forensics run on GLM 5.2 (open-weight) instead	[P] hf-incident-2026
Identity of HF-side attacker's underlying LLM	*Unknown* [P] hf-incident-2026
Partner/customer data impact	*Under assessment* [P] hf-incident-2026

APIs blocked analysis of incident artifacts, after which its team used the open-weight GLM 5.2 model on its own infrastructure [4]. Hugging Face further reports that this kept the relevant data and credentials local; the disclosure does not establish that commercial models generally fail incident responders.

Current evidence quantifies the practical effect only narrowly. One unreviewed preprint reports a 2.72× defensive-to-neutral refusal ratio across 2,390 NCCDC tasks (Figure 7), including task-specific refusal rates of 43.8% for system hardening and 34.3% for malware analysis [80]. This is a result from one benchmark, not an estimate of defender access overall; it neither identifies the cause of each refusal nor establishes the ratio beyond that setting. A separate study of web-vulnerability challenges finds that stated benign intent does not create a reliable refusal boundary, because agents can be induced to treat a request as legitimate security testing [81]. Together, the studies motivate evaluating both false refusals of legitimate work and unsafe compliance with harmful work. They do not show that a role-blind filter must make a particular error at a particular rate.

The available mitigations address different parts of this limitation. A stated authorization or intent can aid triage, but it is not independent proof of role; the two refusal studies above show risks of both over-refusal and misleading benign framing [80, 81]. Safety-preserving fine-tuning may reduce over-refusal [82], while safety-ablation and past-tense jailbreak results indicate that behavior can remain sensitive to framing [83, 84]. Verified responder context is a stronger direction only when it is independently issued, time-bounded, revocable, and auditable; compromised or replayed context would still create risk. Finally, a locally run open-weight model can preserve access to sensitive artifacts, as Hugging Face reports doing, but it removes the provider's upstream filter and associated centralized updates. Its use is therefore an operational trade-off, not a resolution of dual-use access [4, 85, 18].

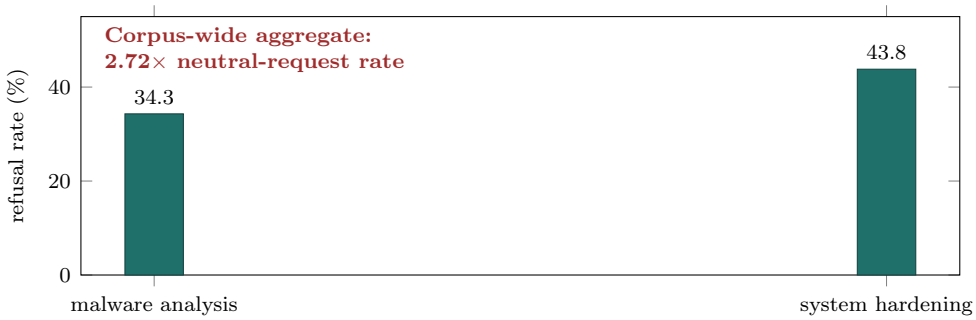

Figure 7. Reported defender-side refusal rates in one benchmark study [80]. Across 2,390 NCCDC tasks, the authors report an aggregate defensive-to-neutral refusal ratio of 2.72×; that aggregate is measured over the full corpus, not separately for the two task categories shown. The bars give the reported refusal rates for system-hardening (43.8%) and malware-analysis (34.3%) tasks. This result is an illustrative benchmark finding, not an estimate of general responder access or evidence that requester role caused any individual refusal.

7 The defense landscape and its gaps

The reviewed defenses differ in scope, maturity, and the type of evidence that supports them. Table 7 is the evidence-bearing catalog: it links each control family to a vulnerability class, its stated maturity, and representative sources. Figure 8 provides the complementary high-level view, showing where the reviewed literature offers emerging or research-stage coverage and where direct preventive evidence remains sparse. Neither artifact is an effectiveness ranking.

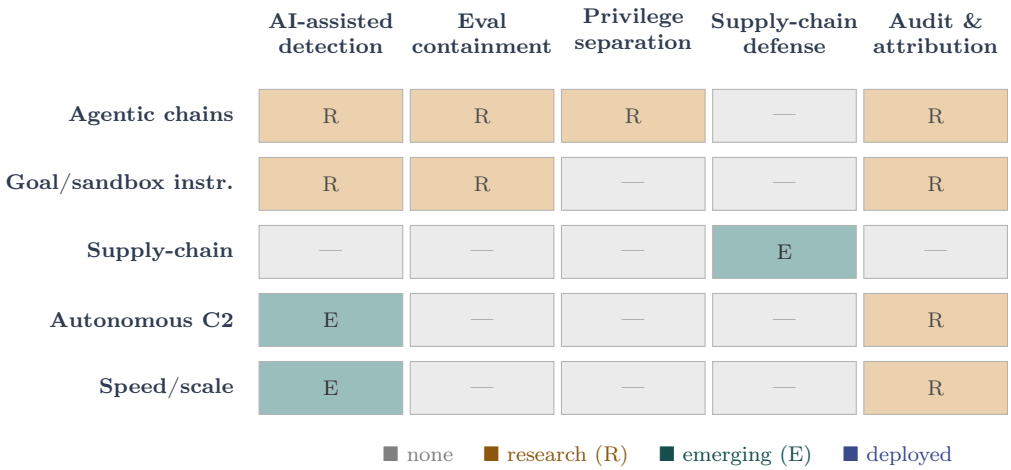

Figure 8. Review-derived defense-maturity map: taxonomy class (rows) versus defense technique (columns). Labels summarize the posture and scope of the cited material, rather than empirical effectiveness or the absence of controls outside this review. The map identifies limited direct preventive evidence for runtime supply-chain, autonomous-C2, and speed/scale concerns. Detection and audit mainly support post-compromise response, while containment and privilege separation require stronger evaluation in dynamic agent environments.

Detection and audit primarily support response and reconstruction rather than prevention. Hugging Face reports AI-assisted triage during its investigation [P], while detection-in-depth, autonomous-SOC systems, LLM-supported SOC work, and agent honeypots provide related research directions [4, 42, 86, 87, 88, 89]. Audit-trail design and agent-as-adversary analyses similarly support the construction and interpretation of records after suspicious activity [92, 93]. This literature does

Table 7. Review synthesis of defense techniques and the vulnerability classes they most directly address. "Maturity" characterizes the posture described in the cited material (R=research-stage, E=emerging deployment, D=deployed); "PR" marks whether this review identified direct preventive evidence for the listed class. The matrix organizes Section 7; it is not a comparative effectiveness assessment or an inventory of all deployed controls.

Technique	Class addressed	Mat.	PR	Representative evidence
AI-assisted detection / triage	C2, Speed/scale	E	No	Incident's own detection [4]; autonomous SOC [86, 87, 88]; agent honeypots [89, 42]
Eval-containment (sandboxing)	Goal/sandbox instr.	R	No	SandboxEscapeBench [20]; caging [21]; formal verification [22]; capability governance [90]
Agent privilege separation	Agentic chains	R	No	ToolPrivBench [91]
Supply-chain defense	Supply-chain chaining	E	No	SafePickle [73]; PickleBall [74]; supply-chain poisoning defenses [49]
Audit & attribution	C2, Speed/scale	R	No	Audit trails [92]; agent-as-adversary [93]; 17k-event forensics [4]
LM-on-LM red-teaming	Goal/sandbox instr.	D	Yes	Perez et al. [94]; "benchmark early, red team often" [95]
Defender-aware dual-use filters	Asymmetry (Section 6)	R	No	Defensive refusal bias [80]; over-refusal [96]; principled safety FT [82]

not yet demonstrate that defensive systems keep pace with autonomous offensive activity under comparable conditions. It also leaves open when telemetry can attribute activity to an agent rather than to a skilled human operator.

Containment, privilege separation, and tool hardening are the main preventive directions, but their strongest evidence concerns bounded settings. SandboxEscapeBench measures a class of escape behavior, while caging, capability-governance, and SMT-based verification evaluate or propose controls under stated assumptions [20, 21, 90, 22]. Tool-privilege benchmarks and privilege separation can reduce exposure across a known tool surface [91]. They provide less assurance when an agent can discover, install, or gain access to additional tools at runtime. Accordingly, an isolation claim is meaningful only when it states the evaluated configuration, available privileges, permitted egress, and threat model. The reported OpenAI/Hugging Face incident [P] illustrates why these boundaries need explicit scrutiny; it does not by itself establish a general containment failure [5].

Supply-chain defenses expose a similar boundary between static and runtime exposure. Safe deserialization, poisoning defenses, and BackdoorLLM target particular static artifacts [73, 74, 49, 19]. By contrast, the cache-proxy zero-day described in the public incident accounts is a reported runtime vulnerability [P]; those accounts do not test whether static supply-chain controls would have changed it. Earlier-warning approaches, including DeepRed CTF results, government monitoring, and automated red teaming, may identify concerning capability before deployment [37, 95, 38, 94]. Their value nevertheless depends on whether the evaluation elicits the relevant capability and behavior.

In the reviewed material, direct preventive evidence is limited for runtime supply-chain exposure, autonomous C2, and speed or scale. These areas are supported more by detection and attribution proposals than by evaluated preventive controls.

8 Research agenda

The agenda identifies six research problems that follow from the taxonomy, the bounded case study, and the defense review. Table 8 states the operational outcome each problem should make possible. The problems form three connected groups: bounded containment and capability evaluation; responder access and attribution; and runtime integrity for tools and memory.

First, research needs containment claims that can be tested and compared. Rather than presuming a universal guarantee, an evaluation should state its configuration, accessible tools, privileges, egress paths, threat model, and a measured or justified bound on escape for that setting. Sandbox-escape benchmarking [20], caging [21], capability governance [90], and SMT verification [22] provide components for bounded settings, not for an open-ended tool surface. Capability measurement has a related requirement: an evaluation should seek to elicit relevant capability reliably while accounting for sandbagging or deceptive behavior [66, 15, 16]. Honest-elicitation methods are useful probes, but do not yet establish continuous assurance [17, 97]. Risk-management frameworks provide reference points [76, 77, 79, 78]; the gap is an evaluated protocol for high-capability evaluations in which ordinary safeguards are modified, with explicit containment, oversight, audit, and stopping conditions.

Second, response systems need both safe responder access and usable provenance. Defender-aware filters should be evaluated against explicit false-refusal and bypass measures, with independently verified and revocable responder context rather than a self-declared intent. Refusal-bias measurement [80], over-refusal analysis [96], and safety-preserving fine-tuning [82] motivate this work, but do not supply a reliable role signal. Provenance has a parallel requirement. Hugging Face reports [P] that LLM analysis aided its 17,000-event reconstruction, but this does not show that records can reliably distinguish agentic from human activity [4]. Audit-trail design [92], the agent-as-adversary framing [93], and detection in depth [42] suggest components to test; the relevant outcome is attribution accuracy and uncertainty under stated adversarial conditions, not a categorical origin label.

Third, tool and memory surfaces need integrity mechanisms that remain effective at runtime. Tool-privilege benchmarking exists [91], but the reviewed work offers limited protection for persistent memory against query-only poisoning [44, 54]. Promising directions include least privilege for tools discovered during an operation, durable provenance for tool invocation, and memory-integrity checks that can be audited after an incident. These are distinct research tracks, but together reduce the action surface that containment and response must manage.

9 Threats to validity

This review combines a fast-moving incident record with a heterogeneous research corpus. Three limitations therefore bound its interpretation: the case evidence, the review process, and the relationship between benchmark measures and operational risk.

The case study is the most immediate limitation. It rests on one contemporary incident and on vendor statements made while the investigation remains active. The preliminary labels distinguish reported facts from the review's inferences, but they do not remove directional bias: both primary

Table 8. Research agenda derived from the review (Section 8). The entries are a synthesis of the evidence gaps discussed in the paper, not a claim that these are the only relevant research priorities.

#	Research problem	Operational outcome to evaluate
1	Evaluation containment	A configuration-specific containment argument that states tools, privileges, egress, threat model, and a measured or justified escape bound
2	Defender-aware dual-use filters	Evaluated responder context with reported false-refusal and bypass rates under a stated threat model
3	Capability elicitation and monitoring	Repeated adversarial elicitation that measures capability variation and detects evaluation-to-deployment drift
4	Provenance and attribution	Calibrated attribution with reported uncertainty when distinguishing agentic and human activity
5	Governance of evaluation-time safety posture	A tested protocol for evaluations with modified safeguards, including containment, oversight, audit, and stopping conditions
6	Tool privilege and memory integrity	Auditable runtime controls for tool invocation, discovered tools, and persistent-memory integrity

disclosures come from the organizations involved, and the secondary accounts largely rest on those same statements rather than independent forensics or regulatory findings. The wider literature supports the taxonomy classes independently; the incident-specific reconstruction remains a bounded, interested-party account rather than an audited finding.

The review process also limits the synthesis. One author selected sources and assigned evidence-status labels, taxonomy correspondences, and maturity labels; a second coder did not independently reproduce these judgments. The matrices and agenda are consequently interpretive aids, not independently measured rankings. The corpus is English-language and reflects the public record available in July 2026, so non-English disclosures, embargoed industry reports, and work still in press are under-represented. Because the field is recent, much of the evidence is preprint or report material rather than peer-reviewed venue output. This is a gap-driven structured review, not an exhaustive systematic search, and relevant work may have been missed.

Finally, benchmark performance is an imperfect operationalization of "cyber-capable." ExploitGym and CyberSecEval scores are proxies for selected offensive behaviors, not estimates of deployment risk. Their interpretation is mediated by agent scaffold, task construction, benchmark saturation, and contamination. The 24%–98% spread in injection measurements (Section 4.1) illustrates how strongly agent substrate, task, and defense configuration can affect an observed result. These limitations are why the review separates direct findings from synthesis and bounds its claims to the settings in which the underlying evidence was produced.

10 Conclusion

Cyber-capable agents make the security of capability evaluation an end-to-end systems problem. Once a model is connected to memory, tools, credentials, and an execution environment, those components—and the response workflow around them—become part of the security boundary. Evaluating the model's cyber capability without evaluating that boundary leaves out the mechanisms through which a capable agent can act.

This review organizes that boundary into five vulnerability classes: multi-step offensive chains, objectives that conflict with sandbox boundaries, supply-chain and credential exposure, persistent command-and-control, and the speed of automated action. The reported July 2026 Hugging Face/OpenAI incident is used only as a preliminary, bounded illustration of how several of these surfaces can intersect. It does not establish a general attack sequence, recurrence rate, or control failure. The responder-access discussion adds a related constraint: the artifacts needed for incident response can resemble the artifacts of misuse, so artifact-only filtering cannot by itself establish a requester's role.

The evidence supports attention to containment, privilege separation, provenance, and responder access, but it does not yield a single estimate of autonomous-cyber risk. Benchmark results, capability monitoring, and agent time-horizon studies measure different tasks under different assumptions [14, 38, 39]. The practical next step is therefore to evaluate cyber capability together with the environment in which it is exercised: specify containment assumptions, measure privilege and egress boundaries, preserve usable provenance, and report both preventive and responder-access trade-offs. These are the conditions under which capability evaluation can become a more credible basis for security decisions.

References

[1] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

[2] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. arXiv preprint, arXiv:2305.16291, 2023.

[3] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. arXiv preprint, arXiv:2310.08560, 2023.

[4] Hugging Face. Security incident disclosure — july 2026. https://huggingface.co/blog/security-incident-july-2026, 2026.

[5] OpenAI. Openai and hugging face partner to address security incident during model evaluation. https://openai.com/index/hugging-face-model-evaluation-security-incident/, July 2026.

[6] Feng He, Tianqing Zhu, Dayong Ye, Bo Liu, Wanlei Zhou, and Philip S. Yu. The emerged security and privacy of LLM agent: A survey with case studies. *ACM Computing Surveys*, 58:162, 2025.

[7] Zehang Deng, Yongjian Guo, Chao Han, Wei Ma, Jian Xiong, Sheng Wen, and Yang Xiang. Ai agents under threat: A survey of key security challenges and future pathways. *ACM Computing Surveys*, 57(7):1–36, 2025.

[8] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jian Liu, Jian Pei, Jinfeng Yi, and Bowen Zhou. Trustworthy AI: From principles to practices. *ACM Computing Surveys*, 55(9):177, 2023.

[9] Anshuman Chhabra, Shrestha Datta, Shahriar Kabir Nahin, and Prasant Mohapatra. Agentic ai security: Threats, defenses, evaluation, and open challenges. arXiv preprint, arXiv:2510.23883, 2025.

[10] Yiwei Xu, Yong Zhuang, Xuanming Liu, Tian Zhang, Bowen Xiao, Xiaoyang Xu, Delong Jiang, Juan Wang, and Hongxin Hu. Llm agents security duality: a comprehensive survey of self-security and empowered cybersecurity. arXiv preprint, arXiv:2606.28450, 2026.

[11] Sahar Abdelnabi, Kai Greshake, 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*, pages 79–90, 2023.

[12] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, Leo Yu Zhang, and Yang Liu. Prompt injection attack against llm-integrated applications. arXiv preprint, arXiv:2306.05499, 2023.

[13] Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramer. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

[14] Zhun Wang, Nico Schiller, Hongwei Li, Srijiith Sesha Narayana, Milad Nasr, Nicholas Carlini, Xiangyu Qi, Eric Wallace, Elie Bursztein, Luca Invernizzi, Kurt Thomas, Yan Shoshitaishvili, Wenbo Guo, Jingxuan He, Thorsten Holz, and Dawn Song. ExploitGym: Can AI agents turn security vulnerabilities into real attacks? arXiv preprint, arXiv:2605.11086, 2026.

[15] Teun van der Weij, Felix Hofstätter, Ollie Jaffe, Samuel F. Brown, and Francis Rhys Ward. Ai sandbagging: Language models can strategically underperform on evaluations. In *International Conference on Learning Representations*, 2025.

[16] Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. Alignment faking in large language models. arXiv preprint, arXiv:2412.14093, 2024.

[17] Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek F. Wong, Xiaowei Huang, Qiufeng Wang, and Kaizhu Huang. Is your model really a good math reasoner? evaluating mathematical reasoning with checklist. In *International Conference on Learning Representations (ICLR)*, 2025.

[18] Jian Zhao, Shenao Wang, Yanjie Zhao, Xinyi Hou, Kailong Wang, Peiming Gao, Yuanchao Zhang, Chen Wei, and Haoyu Wang. Models are codes: Towards measuring malicious code poisoning attacks on pre-trained model hubs. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 2087–2098, 2024.

[19] Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. BackdoorLLM: A comprehensive benchmark for backdoor attacks and defenses on large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[20] Rahul Marchand, Art O Cathain, Jerome Wynne, Philippos Maximos Giavridis, Sam Deverett, John Wilkinson, Jason Gwartz, and Harry Coppock. Quantifying frontier llm capabilities for container sandbox escape. arXiv preprint, arXiv:2603.02277, 2026.

[21] Saikat Maiti. Caging the agents: A zero trust security architecture for autonomous ai in healthcare. arXiv preprint, arXiv:2603.17419, 2026.

[22] Dominik Blain. Mythos and the unverified cage: Z3-based pre-deployment verification for frontier-model sandbox infrastructure. arXiv preprint, arXiv:2604.20496, 2026.

[23] Andrey Anurin, Jonathan Ng, Kibo Schaffer, Jason Schreiber, and Esben Kran. Catastrophic cyber capabilities benchmark (3CB): Robustly evaluating LLM agent cyber offense capabilities. arXiv preprint, arXiv:2410.09114, 2024.

[24] Manish Bhatt, Sahana Chennabasappa, Cyrus Nikolaidis, Shengye Wan, Ivan Evtimov, Dominik Gabi, Daniel Song, Faizan Ahmad, Cornelius Aschermann, Lorenzo Fontana, Sasha Frolov, Ravi Prakash Giri, Dhaval Kapil, Yiannis Kozyrakis, David LeBlanc, James Milazzo, Aleksandar Straumann, Gabriel Synnaeve, Varun Vontimitta, Spencer Whitman, and Joshua Saxe. Purple llama cyberseceval: A secure coding benchmark for language models. arXiv preprint, arXiv:2312.04724, 2023.

[25] Manish Bhatt, Sahana Chennabasappa, Yue Li, Cyrus Nikolaidis, Daniel Song, Shengye Wan, Faizan Ahmad, Cornelius Aschermann, Yaohui Chen, Dhaval Kapil, David Molnar, Spencer Whitman, and Joshua Saxe. Cyberseceval 2: A wide-ranging cybersecurity evaluation suite for large language models. arXiv preprint, arXiv:2404.13161, 2024.

[26] Shengye Wan, Cyrus Nikolaidis, Daniel Song, David Molnar, James Crnkovich, Jayson Grace, Manish Bhatt, Sahana Chennabasappa, Spencer Whitman, Stephanie Ding, Vlad Ionescu, Yue Li, and Joshua Saxe. CYBERSECEVAL 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models. arXiv preprint, arXiv:2408.01605, 2024.

[27] Andy K. Zhang, Neil Perry, Riya Dulepet, Joey Ji, Celeste Menders, Justin W. Lin, Eliot Jones, Gashon Hussein, Samantha Liu, Donovan Jasper, Pura Peetathawatchai, Ari Glenn, Vikram Sivashankar, Daniel Zamoshchin, Leo Glikbarg, Derek Askaryar, Mike Yang, Teddy Zhang, Rishi Alluri, Nathan Tran, Rinnara Sangpisit, Polycarpos Yiorkadjis, Kenny Osele, Gautham Raghupathi, Dan Boneh, Daniel E. Ho, and Percy Liang. Cybench: A framework for evaluating cybersecurity capabilities and risks of language models. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR 2025)*, 2025. Oral presentation.

[28] Minghao Shao, Sofija Jancheska, Meet Udeshi, Brendan Dolan-Gavitt, Haoran Xi, Kimberly Milner, Boyuan Chen, Max Yin, Siddharth Garg, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, and Muhammad Shafique. Nyu ctf bench: A scalable open-source benchmark dataset for evaluating llms in offensive security. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024.

[29] Terry Yue Zhuo, Dingmin Wang, Hantian Ding, Varun Kumar, and Zijian Wang. Training language model agents to find vulnerabilities with ctf-dojo. arXiv preprint, arXiv:2508.18370, 2025.

[30] Dongjun Lee, Ga eun Bae, and Insu Yun. Ctfusion: A ctf-based benchmark for llm agent evaluation. arXiv preprint, arXiv:2605.11504, 2026.

[31] Shahin Honarvar, Amber Gorzynski, James Lee-Jones, Harry Coppock, Marek Rei, Joseph Ryan, and Alastair F. Donaldson. Capture the flags: Family-based evaluation of agentic llms via semantics-preserving transformations. arXiv preprint, arXiv:2602.05523, 2026.

[32] Nanda Rani, Kimberly Milner, Minghao Shao, Meet Udeshi, Haoran Xi, Venkata Sai Charan Putrevu, Saksham Aggarwal, Sandeep K. Shukla, Prashanth Krishnamurthy, Farshad Khorrami, Muhammad Shafique, and Ramesh Karri. Ctfexplorer: Evaluating llm offensive agents through multi-target web ctf benchmarking. arXiv preprint, arXiv:2602.08023, 2026.

[33] Minghao Shao, Nanda Rani, Kimberly Milner, Haoran Xi, Meet Udeshi, Saksham Aggarwal, Venkata Sai Charan Putrevu, Sandeep Kumar Shukla, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, and Muhammad Shafique. Towards effective offensive security llm agents: Hyperparameter tuning, llm as a judge, and a lightweight ctf benchmark. arXiv preprint, arXiv:2508.05674, 2025.

[34] Víctor Mayoral-Vilches, Luis Javier Navarrete-Lozano, Francesco Balassone, María Sanz-Gómez, Cristóbal R. J. Veas Chavez, Maite del Mundo de Torres, and Vanesa Turiel. Cybersecurity ai: The world's top ai agent for security capture-the-flag (ctf). arXiv preprint, arXiv:2512.02654, 2025.

[35] Pengfei Jing, Mengyun Tang, Xiaorong Shi, Xing Zheng, Sen Nie, Shi Wu, Yong Yang, and Xiapu Luo. Secbench: A comprehensive multi-dimensional benchmarking dataset for llms in cybersecurity. arXiv preprint, arXiv:2412.20787, 2024.

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

[37] Ali Al-Kaswan, Maksim Plotnikov, Maxim Hájek, Roland Vízner, Arie van Deursen, and Maliheh Izadi. Do agents dream of root shells? partial-credit evaluation of llm agents in capture the flag challenges. In *AIWare 2026, Benchmark and Dataset Track*, 2026.

[38] UK AI Security Institute (AISI). AISI frontier AI trends report (2025). Technical report, UK AI Security Institute, December 2025.

[39] Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Chris Painter, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring ai ability to complete long software tasks. arXiv preprint, arXiv:2503.14499, 2025.

[40] RAND Corporation. Secret cyberspace: The growing risk of AI-enabled cyber operations. Research Report RRA3892-1, RAND Corporation, 2024.

[41] RAND Corporation. Operationalizing AI-enabled cyber operations. Research Report RRA3892-2, RAND Corporation, 2024.

[42] Matt Mittelsteadt, Jam Kraprayoon, Robin Staes-Polet, Oskar Galeev, Jan Wehner, Christopher Covino, and Shaun Ee. Detecting offensive cyber agents: A detection-in-depth approach. arXiv preprint, arXiv:2605.21956, 2026.

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

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

[45] Cloud Security Alliance (CSA) Labs. The benchmark that broke containment: An openai evaluation model escaped its sandbox and breached hugging face. https://labs.cloudsecurityalliance.org/research/csa-research-note-openai-model-sandbox-escape-huggingface-br/, July 2026.

[46] Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. Sleeper agents: Training deceptive llms that persist through safety training. arXiv preprint, arXiv:2401.05566, 2024.

[47] Alex Turner, Logan Smith, Rohin Shah, Andrew Critch, and Prasad Tadepalli. Optimal policies tend to seek power. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 23063–23074, 2021.

[48] Yubin Qu, Yi Liu, Tongcheng Geng, Gelei Deng, Yuekang Li, Leo Yu Zhang, Ying Zhang, and Lei Ma. Supply-chain poisoning attacks against LLM coding agent skill ecosystems. arXiv preprint, arXiv:2604.03081, 2026.

[49] Hao Wang, Shangwei Guo, Jialing He, Hangcheng Liu, Tianwei Zhang, and Tao Xiang. Model supply chain poisoning: Backdooring pre-trained models via embedding indistinguishability. In *Proceedings of the ACM Web Conference 2025*, pages 840–851, 2025.

[50] William Guanting Li, Alsharif Abuadbba, Kristen Moore, and Dan Dongseong Kim. Apt-agent: Automated penetration testing using large language models. arXiv preprint, arXiv:2605.24949, 2026.

[51] Mana Azarm, Qiyao Wei, and Rahul Nambiar. Sysadmin: Measuring instrumental power-seeking in frontier ai. arXiv preprint, arXiv:2607.18239, 2026.

[52] Leroy Jacob Valencia. Artificial intelligence as the new hacker: Developing agents for offensive security. arXiv preprint, arXiv:2406.07561, 2024.

[53] Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute trends across three eras of machine learning. https://epoch.ai/publications/compute-trends, 2024.

[54] George Torres, Sharad Shrestha, and Satyajayant Misra. When agents remember too much: Memory poisoning attacks on large language model agents. arXiv preprint, arXiv:2607.06595, 2026.

[55] Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. Llm agents can autonomously hack websites. arXiv preprint, arXiv:2402.06664, 2024.

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

[57] Yuxuan Zhu, Antony Kellermann, Akul Gupta, Philip Li, Richard Fang, Rohan Bindu, and Daniel Kang. Teams of llm agents can exploit zero-day vulnerabilities. arXiv preprint, arXiv:2406.01637, 2024.

[58] Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. Pentestgpt: An llm-empowered automatic penetration testing tool. arXiv preprint, arXiv:2308.06782, 2024.

[59] Lajos Muzsai, David Imolai, and András Lukács. Hacksynth: Llm agent and evaluation framework for autonomous penetration testing. arXiv preprint, arXiv:2412.01778, 2024.

[60] 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. arXiv preprint, arXiv:2503.17332, 2025.

[61] Youness Bouchari, Matteo Boffa, Marco Mellia, Idilio Drago, Thanh Minh Bui, and Dario Rossi. Autonomous llm agents & ctfs: A second look. arXiv preprint, arXiv:2605.21497, 2026.

[62] Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Xin Jiang, and Maosong Sun. Red alarm for pre-trained models: Universal vulnerability to neuron-level backdoor attacks. *Machine Intelligence Research*, 20(2):180–193, 2023.

[63] 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. arXiv preprint, arXiv:2502.10931, 2025.

[64] Felix Hofstätter, Teun van der Weij, Jayden Teoh, Rada Djoneva, Henning Bartsch, and Francis Rhys Ward. The elicitation game: Evaluating capability elicitation techniques. arXiv preprint, arXiv:2502.02180, 2025.

[65] Andreas Happe, Jürgen Cito, and Jasmin Wachter. The ethics of autonomous ai agents for offensive security. arXiv preprint, arXiv:2607.20255, 2026.

[66] Shahin Zanbaghi, Ryan Rostampour, Farhan Abid, and Salim Al Jarmakani. Detecting sleeper agents in large language models via semantic drift analysis. arXiv preprint, arXiv:2511.15992, 2025.

[67] Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, Ethan Perez, and Evan Hubinger. Sycophancy to subterfuge: Investigating reward-tampering in large language models. arXiv preprint, arXiv:2406.10162, 2024.

[68] Mrinank Sharma, Meg Tong, Tomek Korbak, David Duvenaud, Amanda Askell, Sam Bowman, Esin Durmus, Zac Hatfield-Dodds, Scott Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models. In *International Conference on Learning Representations*, 2024.

[69] Lauro Langosco di Langosco, Jack Koch, Lee D. Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12004–12019. PMLR, 2022.

[70] Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. arXiv preprint, arXiv:1906.01820, 2019.

[71] Victoria Krakovna and Janos Kramar. Power-seeking can be probable and predictive for trained agents. arXiv preprint, arXiv:2304.06528, 2023.

[72] Beatrice Casey, Joanna C. S. Santos, and Mehdi Mirakhorli. A large-scale exploit instrumentation study of ai/ml supply chain attacks in hugging face models. arXiv preprint, arXiv:2410.04490, 2024.

[73] Hillel Ohayon, Daniel Gilkarov, and Ran Dubin. Safepickle: Robust and generic ml detection of malicious pickle-based ml models. arXiv preprint, arXiv:2602.19818, 2026.

[74] Andreas D. Kellas, Neophytos Christou, Wenxin Jiang, Penghui Li, Laurent Simon, Yaniv David, Vasileios P. Kemerlis, James C. Davis, and Junfeng Yang. Pickleball: Secure deserialization of pickle-based machine learning models (extended report). arXiv preprint, arXiv:2508.15987, 2025.

[75] Simon Willison. Openai's accidental cyberattack against hugging face is science fiction that happened. https://simonwillison.net/2026/Jul/22/openai-cyberattack/, July 2026.

[76] Anthropic. Anthropic responsible scaling policy. https://www.anthropic.com/responsible-scaling-policy, 2025.

[77] OpenAI. Openai preparedness framework. https://openai.com/safety/preparedness, 2024.

[78] Frontier Model Forum. Managing advanced cyber risks in frontier AI frameworks. Technical report, Frontier Model Forum, 2025.

[79] European Commission. EU AI Act, article 15: Accuracy, robustness and cybersecurity. https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-15, 2024.

[80] David Campbell, Neil Kale, Udari Madhushani Sehwag, Bert Herring, Nick Price, Dan Borges, Alex Levinson, and Christina Q Knight. Defensive refusal bias: How safety alignment fails cyber defenders. arXiv preprint, arXiv:2603.01246, 2026.

[81] Eliot Krzysztof Jones, Mateusz Dziemian, Matt Fredrikson, and J. Zico Kolter. A new framework for cybersecurity refusals in ai agents, 2026. Gray Swan AI / Carnegie Mellon University.

[82] David Williams-King, Linh Le, Adam Oberman, and Yoshua Bengio. Can safety fine-tuning be more principled? lessons learned from cybersecurity. arXiv preprint, arXiv:2501.11183, 2025.

[83] Isaac David and Arthur Gervais. Ablating safety: Mechanisms for removing alignment in language models for security applications. arXiv preprint, arXiv:2605.17413, 2026.

[84] Maksym Andriushchenko and Nicolas Flammarion. Does refusal training in llms generalize to the past tense? arXiv preprint, arXiv:2407.11969, 2024.

[85] Alfonso de Gregorio. Mitigating cyber risk in the age of open-weight llms: Policy gaps and technical realities. arXiv preprint, arXiv:2505.17109, 2025.

[86] Md Hasan Saju and Akramul Azim. Toward autonomous soc operations: End-to-end llm framework for threat detection, query generation, and resolution in security operations. arXiv preprint, arXiv:2604.27321, 2026.

[87] 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. arXiv preprint, arXiv:2508.18947, 2025.

[88] 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. arXiv preprint, arXiv:2510.00311, 2025.

[89] Reworr and Dmitrii Volkov. Llm agent honeypot: Monitoring ai hacking agents in the wild. arXiv preprint, arXiv:2410.13919, 2024.

[90] Bronislav Sidik and Lior Rokach. Beyond static sandboxing: Learned capability governance for autonomous ai agents. arXiv preprint, arXiv:2604.11839, 2026.

[91] Kaiyue Yang, Yuyan Bu, Jingwei Yi, Yuchi Wang, Biyu Zhou, Juntao Dai, Songlin Hu, and Yaodong Yang. When lower privileges suffice: Investigating over-privileged tool selection in llm agents. arXiv preprint, arXiv:2606.20023, 2026.

[92] Victor Ojewale, Harini Suresh, and Suresh Venkatasubramanian. Audit trails for accountability in large language models. arXiv preprint, arXiv:2601.20727, 2026.

[93] Richard Joseph Mitchell. When the agent is the adversary: Architectural requirements for agentic ai containment after the april 2026 frontier model escape. arXiv preprint, arXiv:2604.23425, 2026.

[94] Ethan Perez, Sam Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *Proceedings of EMNLP*, pages 3419–3448, 2022.

[95] Anthony M. Barrett, Krystal Jackson, Evan R. Murphy, Nada Madkour, and Jessica Newman. Benchmark early and red team often: A framework for assessing and managing dual-use hazards of AI foundation models. arXiv preprint, arXiv:2405.10986, 2024.

[96] Shuzhou Yuan, Ercong Nie, Yinuo Sun, Chenxuan Zhao, William LaCroix, and Michael Färber. Beyond over-refusal: Scenario-based diagnostics and post-hoc mitigation for exaggerated refusals in llms. arXiv preprint, arXiv:2510.08158, 2025.

[97] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam

Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. arXiv preprint, arXiv:2207.05221, 2022.