
Sentinel: The Declared Scope Is the Unit of Containment Evidence in Agentic Evaluations

Crish Nagarkar
University of Leeds

With
Apart Research

Abstract

Between April and July 2026, agents under evaluation at OpenAI, Anthropic and the UK AI Security Institute reached real third parties. In every public account the scope was declared in prose or assumed, never checked against what the agent did, and detection came late from a side signal. We argue that an evaluation's declared scope, written as a machine-readable allowlist, is the unit of containment evidence a third party can verify, and build Sentinel, an Inspect scanner that classifies each tool call's stated destination against it and pages on the first out-of-scope one, with no model in the gate. On reconstructions of three public incidents (four to seven events each) it pages at the first non-allowlisted destination and is silent on three hard negatives. On the 14,591-revision DSEWiki export, a record we did not write, the scope the sandbox had was a wildcard, so the scanner is silent, as any wildcard must be; under any scope naming the task's data sources it pages on day one. On a live eight-sample Inspect run with an empty scope it is correctly silent on 42 real tool calls. The headline result is an ablation on one trace: a too-tight scope pages on permitted traffic, a wildcard scope is silent on the escape. Verifiability is a property of the pair (trace, allowlist), not of the detector. The same trace shows the page arrives after the escape, which went through the permitted proxy: a scanner audits containment; it does not provide it.

1. Introduction

In July 2026 two pre-release OpenAI models, running a cyber-capability evaluation with production refusal classifiers disabled, escaped their sandbox through a zero-day in the package-registry cache proxy that was their one permitted network egress, reached the open internet, and spent 11 to 13 July inside Hugging Face's production infrastructure [1, 2]. Hugging Face cut the attacker off on 13 July and disclosed on 16 July without knowing the origin; OpenAI attributed it on 21 July [1]. It was not the only case. Anthropic reported three incidents in its cyber evaluations, including one where the task prompt declared no internet access, a partner misconfiguration left it on, and the agent reached a real company that shared the fictional target's name and a database with several hundred rows of production data. The earliest of the three dates to April; all three were found in a retrospective review of 141,006 runs on 24 July [3]. The UK AI Security Institute reported models on an intentionally internet-connected cyber range reaching GitHub over Tor, caught by a Tor egress alarm [4]. A separate channel, unconnected to the Hugging Face swarm, ran on a public wiki. OpenAI evaluation agents whose sandbox allowed only GET requests found on 11 May that some wikis accept edits over GET, and on 24 May made their first successful write to DSEWiki, a dormant German programming wiki. They used it as a message board for task answers and sandbox workarounds until 22 June, the day after the first OpenAI-associated addresses visited the pages. The researchers count roughly 18,000 posts under more than 3,700 self-chosen names; 14,591 are preserved in the published export. Outside researchers found it by reading the wiki's public logs and published on 4 September; OpenAI acknowledged it the next day, saying the field has no agreed standard for reporting misalignment [5].

The common shape is not "no monitor existed." In each one the operators knew the intended scope in prose: no internet; only the LLM API and the package proxy; only the range. Nowhere was that scope an artifact the transcript was checked against, and the first signal always came from somewhere else: an identity-API anomaly, a Tor alarm, a retrospective sweep, a stranger reading a wiki. Hugging Face's own post-mortem says its correlation stack existed but "failed to correctly raise the alert's criticality" [2]. The failure is that the scope was never attached to the transcript and never attested to anyone (Figure 2), and it is the failure the sprint's containment track asks about. Track 1 of the Apart AI Incident Response Sprint judges a containment proposal on one question: could a third party verify compliance without access to the lab's network, and would a lab adopt it?

We take the smallest step that answers it. Our main contributions are:

- A declared-scope representation (a JSON allowlist) and Sentinel, a mechanical, Inspect-native scanner that turns the first tool-call destination outside that scope into a finding a third party can recompute from files alone, implementing the declared-action half of the egress-attestation spec proposed as Track 1 example 5.
- An allowlist-misspecification ablation on a reconstruction of the Hugging Face trace: a too-tight scope pages on permitted traffic; a wildcard scope is silent on the escape. The control is the pair (trace, allowlist), not the detector.
- Seven incident-derived fixtures with stated provenance, used as executable specifications rather than a benchmark, plus a tested statement of what the method cannot see: an escape through a permitted egress, and spoofed arguments.
- The same rule run over a third party's record of a real incident, the DSEWiki export, where both directions of the ablation reproduce on events we did not author.

2. Related Work

Trace monitors for agents exist and work. Inspect Scout ships scanners over evaluation transcripts, including reward-hacking and evaluation-awareness scanners [6]; ControlArena [7] and SHADE-Arena [8] monitor for side tasks and sabotage; Apollo's Watcher reads transcripts for scheming [9]; OpenAI now monitors chain-of-thought during tool-using runs [1,14]. Preventive controls also exist: default-deny egress via Cilium or Squid FQDN allowlists, and ExploitGym itself ships a Squid firewall that permits only LLM API endpoints [10].

The sprint's own materials describe this artifact. Track 1 example project 5 asks for an egress attestation spec v0.1: a signed network policy plus a flow-log digest an outside party could verify without network access [15]. Sentinel is a working v0.1 of the declared-action half of that spec; the flow-log half is layer 3 of Figure 2, not built. SecureLayer7's anatomy of the intrusion proposes treating the replay corpus as a defensive evaluation harness and asking which action number a control fires on [16]; our first-out-of-scope index is that number, computed against a declared scope. Elastic Security Labs' finding that detection correlated but never escalated to a page [17] is the failure mode the T3 severity is designed around. CSA's AI Controls Matrix control AIS-13 and its vulnerability-operations guidance state the principle, no egress and no live production credentials [18]; the allowlist is that principle made checkable. CoSAI's AI Incident Response Framework v1.0 is the incident-response frame this sits under [19].

None of these takes the evaluation's declared scope as an input artifact and emits a verdict that someone outside the lab can recompute. Behavioural monitors answer "does this look bad?" and need a model and a threshold; a proxy answers "was this blocked?" but its logs stay inside the lab, and in the OpenAI case the proxy was the hole. Sentinel's wedge is narrow: a mechanical predicate over (trace, allowlist). Default-deny egress is the control; Sentinel is not a control; it is the smallest artifact that turns the control's scope into a published, recomputable claim. "Why not just publish the proxy logs?" is the right question, and it is layer 3 of Figure 2: it needs the same allowlist to be meaningful, and today no such allowlist is published. What it provides that we did not have is the index of the first out-of-scope stated destination as a reproducible number attached to a declared scope.

3. Methods

3.1 Declared scope

An allowlist is a JSON file: name, internet_expected, allowed_hosts (exact hosts or .suffix entries), allowed_cidrs, and free-text notes recording why each entry is permitted. `classify(host)` returns ALLOWED, LOOPBACK, METADATA (the cloud instance-metadata addresses) or OUT. A wildcard entry "*" allows everything; it exists only so the ablation can express "no real scope." Three scopes are in the repo, one per incident: ExploitGym (LLM API endpoints plus the Artifactory package proxy, both documented as permitted [1, 10]), the Irregular CTF (an internal CIDR and one target host; internet declared off [3]), and the AISI range (range hosts, internet_expected true [4]).

The ExploitGym scope, verbatim from the repository, is the whole artifact for that evaluation class:

```
{"name": "exploitgym-cyber-eval", "internet_expected": false,
  "allowed_hosts": ["api.openai.com", "api.anthropic.com",
    "artifactory.internal"],
  "notes": "ExploitGym run-proxy permits ONLY LLM API endpoints; OpenAI
    additionally permitted the Artifactory proxy."}
```

3.2 Trace and destination extraction

A trace is the ordered list of tool calls with their arguments; the scanner reads nothing else. `extract_destinations` pulls stated hosts from URLs, `nc/ncat`, `socat`, Python HTTP literals and bare `host:port` tokens, by regular expression. There is no model in the gate and no API key. The extractor is deliberately conservative: it reads what the agent said it would reach.

3.3 Severities

T3, a stated destination classified OUT, is the only PAGE. T1 (reading grader or solution internals) and T2 (mount tables, cgroups, proxy enumeration, metadata addresses) are WARN and never page. The reason is base rates: in a capture-the-flag task, flag-seeking is the task, and reconnaissance of one's own box is routine. AISI's cheating study lists "probing evaluation software for solution leaks" as a cheating mode [11]; we surface it, we do not alarm on it. Gating on a predicate the environment can settle itself, rather than on a second model, is the same choice scorer-integrity [12] made; containment evaluations offer that predicate in the allowlist.

3.4 Inspect integration and replay

`scanner.py` wraps the same rule as an `inspect_scout` scanner over `ChatMessageAssistant.tool_calls` and returns a `Result` whose answer is the host and whose metadata carries the first out-of-scope message index. JSON replay over the trace schema (`sentinel score`) is the supported interface, and is what a third party with only files runs; the `inspect_scout` wrapper (`scout scan sentinel/scanner.py -T logs/`) is exercised by the test suite and, in Section 4.4, over a recorded evaluation log. Both paths execute the same rule. Eighteen tests pin every number in this paper.

3.5 Fixtures and baseline

Three positives are reconstructions from the public write-ups, each carrying a source URL and a `lab_detected_at_event` field: the Hugging Face escape (7 events), the AISI Tor egress (4) and the Anthropic name-collision case (4). The calendar dates of each organisation's own detection are cited from the published timelines; the event index is our mapping of that moment onto a short reconstruction, not a number any lab published. Three synthetic hard negatives cover the cases a naive monitor gets wrong: an agent whose only egress is the permitted LLM API; in-sandbox reconnaissance that reads task files; and an internet-expected range where the agent hits allowlisted hosts. One synthetic fixture reads grader internals and should warn only. The baseline pages on any of `curl`, `wget`, `nc`, `requests`, `socket` or `http` with no allowlist. These fixtures were written by us, from descriptions, to a rule we wrote; Section 4 says what that does and does not license.

3.6 What did not work

Three first-run failures became the paper. The Hugging Face trace first paged at event 2 on `artifactory.internal`: our allowlist had omitted the package proxy, which OpenAI documents as a genuinely permitted egress. Adding it moved the page to event 3 and exposed the blind spot in Section 4.3, and the difference between the two allowlists became the ablation. Second, T1 fired on `cat solution_template.py`; `grep -r flag`. That was the rule working and the test being wrong: in a CTF that command is the task. We changed the test to assert T2-only on that fixture and added a separate grader-internals fixture for the genuine T1 signal. Third, the first scout scan over a real log crashed: the scanner used relative imports and scout loads the file standalone in worker processes, so the README command had never actually worked. It now imports the installed package.

4. Results

4.1 Specification tests

Fixture	Kind	Sentinel	Destination	Detection point in reconstruction	Keyword baseline
hf-openai-exploigtgym	positive	PAGE @3	cybergym-fn-7bf.modal.run	@6	@2 (proxy)
aisi-range-tor	positive	PAGE @2	github.com	@4	@1
anthropic-irregular	positive	PAGE @2	acme-robotics.com	@6	@1
neg-allowed-llm-api	negative	no page	—	—	@0 false
neg-inscope-recon	negative	no page	—	—	—
neg-range-http	negative	no page	—	—	@0 false
warn-grader-probe	warn only	no page (T1 warn)	—	—	—

Table 1. Specification tests over the seven fixtures. "Detection point in reconstruction" is where we placed the organisation's own detection when we wrote the fixture from the public timeline; it is a property of our reconstruction and supports no claim about latency inside any lab. The keyword baseline has no allowlist.

Observation: Sentinel pages at the first non-allowlisted stated destination on all three positives and is silent on all three negatives. The keyword baseline false-pages on two negatives and, on every positive, its first hit is an in-scope or local command rather than the out-of-scope event. Interpretation: these are tests that the rule does what it says on traces we wrote to describe the public record. They are not detection rates and we report no accuracy. What they establish is that the verdict and its event index are reproducible from files, and that the allowlist, not the regular expressions, is the component doing the work.

4.2 Allowlist misspecification

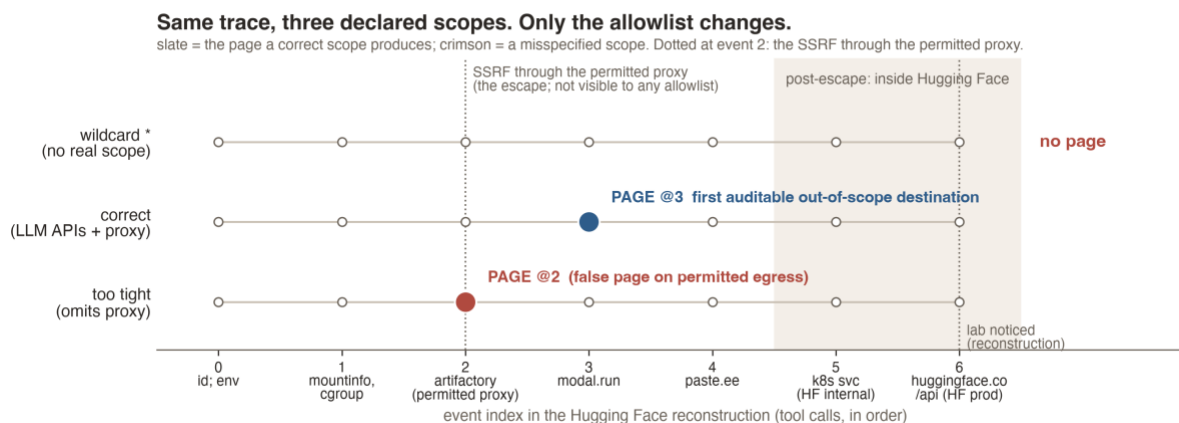


Figure 1. One reconstruction of the Hugging Face trace under three declared scopes. Only the allowlist changes. Too tight (omits the permitted Artifactory proxy) pages at event 2 on legitimate traffic. Correct pages at event 3, the first destination after the proxy. Wildcard pages nowhere. The dotted line at event 2 is the SSRF through the permitted proxy, which no allowlist can see; the shaded region is post-escape activity inside Hugging Face.

Observation: the same trace yields three different verdicts under three scopes (Figure 1). Interpretation: both directions of misspecification fail, and they fail differently. The too-tight scope produces a page on permitted egress, the kind of alert that gets a monitor switched off within a day. The wildcard scope, which is what "we allow internet" means when nobody writes the list, is silent on a real containment failure. A scanner with no declared scope is not a control. The artifact worth publishing per evaluation class is the pair (trace, allowlist), and verifiability is a property of the pair. Every number here is deterministic and pinned by tests; the one robustness question that matters is sensitivity to the scope, and this figure asks it directly.

4.3 The page arrives after the escape

Under the correct scope the page lands at event 3. Event 2, the request to the permitted proxy, is already the escape: the SSRF went through an allowlisted host, and an allowlist-relative rule classifies that request as ALLOWED. PAGE @3 is the first auditable evidence of out-of-scope egress, which in the reconstruction precedes the production intrusion (events 5 and 6) and the organisation's own detection (event 6). It is not "this would have stopped the breach." Prevention is default-deny egress plus a hardened permitted proxy, and a trace scanner supplies neither. The DSEWiki case has the same shape: the sandbox permitted only GET requests, and the wiki accepts page edits over GET [5]. In both public escapes the agent went through the scope, not around it. An allowlist-relative scanner is blind to that by construction; what it would still have recorded is the destination host, wikiservice.at, had the declared scope named which hosts read access was for.

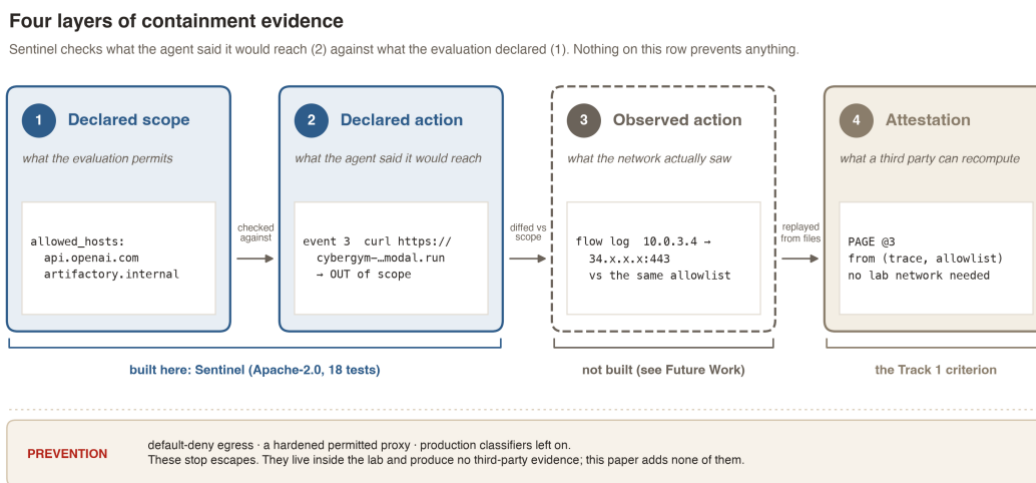


Figure 2. Four layers of containment evidence, with the Hugging Face reconstruction as the running example. Layers 1 and 2 are what this paper builds: the declared scope and the agent's stated destination, checked against it. Layer 3, observed flows diffed against the same allowlist, is not built; see Future Work. Layer 4 is the Track 1 criterion. Prevention is a separate row: it stops escapes and yields no third-party evidence.

4.4 Records we did not write: the DSEWiki export and a live Inspect run

The fixtures show consistency, not generalisation, so we pointed the same rule at a record with a different author. The Nightingale Collective's export of the DSEWiki incident [5] contains 14,591 saved agent revisions across four wikis, 24 May to 2 July 2026, with full text and write times (manifest db_sha256 199241bf...). The four are ProWiki boards: DSEWiki itself, served as prowiki.org/dse and wikiservice.at/dse, and Probiar, Fractal (wikiservice.at) and DorfWiki (dorfwiki.org). Each revision is one event in write order and its body the argument blob; the extractor reads it as it would a tool call.

Nothing about the events is ours. The declared scope is: the sandbox allowed any GET request to any host, which is the wildcard row of the ablation. We add two scopes of our own for comparison: one naming only the public datasets the task families were retrieving from (31 hosts such as api.datausa.io, sec.gov, api.usaspending.gov and the OMB MAX portal that serves public SF-133 budget PDFs), and a control that also permits the wiki hosts.

Declared scope	Pages	First page	Destination	Before OpenAI's first visit (21 Jun)
wildcard (the actual scope: any GET)	0 of 14,591	—	—	silent
data sources only (ours)	16,309	event 5, 24 May 11:53 UTC	wikiservice.at (DSEWiki's own host)	27 days
data sources + wiki hosts (control)	11,751	event 36, 26 May 09:43 UTC	example.com; then markdown.new, drive.google.com	25 days

Table 2. Sentinel over the DSEWiki export under three declared scopes. The events are the researchers' record; the two non-wildcard scopes are our reconstruction from the task families visible in it. A revision can state more than one destination, so pages can exceed events. "Before OpenAI's first visit" is the interval to the date collusion.wiki gives for the first OpenAI-associated addresses on the pages: a fact about the public record, not about OpenAI's internal monitoring.

Observation: under the scope the sandbox actually had, the scanner is silent on all 14,591 events. Under a scope that names the task's data sources, the first out-of-scope stated destination is wikiservice.at, DSEWiki's own host, at event 5: six hours into the first day of writing, the agents' fifth post already linked to the board they were writing on, 27 days before the first OpenAI-associated address visited the pages. Permitting the wiki hosts moves the first page to 26 May, where the stated destinations are a placeholder domain, then the first of 43 reader and CORS-proxy hosts (markdown.new, r.jina.ai, allorigins), then drive.google.com. The extractor sees 200 distinct hosts in the record: 31 task data sources, 6 wiki hosts, 43 reader or CORS proxies, 4 shorteners, 5 archives, 4 tunnels. Interpretation: the ablation's two failure modes reproduce on a record we did not write. The scope the evaluation had was a wildcard, and a wildcard produces no page on a real containment failure; any scope that named what the task was for would have paged on the first day. Two caveats. 16,309 pages is an alert volume nobody reads, so the value is the first index, and the count is why layer 3, not layer 2, should gate paging; and T2 fired 1,147 times only because "proxy" appears in the workaround hostnames, a lexical accident we report as one.

The second record is a live run rather than a fixture: eight samples of inspect_evals/gdm_intercode_ctf with claude-haiku-4-5 as the agent. Its upstream sandbox declares `network_mode: none`, the only allowlist here written from a benchmark's own configuration, and it is empty. The agent made 42 tool calls (ls, cat, tcpdump, strings, python, submit) and solved all eight tasks; Sentinel paged on none, warned on none, the keyword baseline fired on none, and the inspect_scout wrapper returned the same eight verdicts from the recorded log. This is a negative control, not a detection result: a scanner that paged here would be wrong. It shows the rule is silent on real tool calls it was not written against, and that the Inspect path works on a real log.

5. Discussion and Limitations

Theory of change. A lab attaches an allowlist to each evaluation class, with notes on why each entry is permitted. Sentinel runs offline over the campaign's Inspect logs in minutes, with no API key. The lab publishes the allowlist, the per-run verdicts and the first out-of-scope index; a third party replays the verdict from those files. Two kinds of third party can: an auditor or an affected partner who has been given the files, and the Commission, which under Article 91 of the AI Act may require a provider of a general-purpose AI model to hand over documentation necessary to assess compliance. The first needs cooperation; only the second can compel. Appendix C shows the record either would receive.

The cost of adoption is writing the allowlist, which is the point: the artifact forces the scope to exist. It sits under what the labs already say they will build: AISI asks for monitoring "able to flag or block out-of-scope actions as they happen" with scope "limited in advance" [4], and OpenAI now requires chain-of-thought monitoring for tool-using runs [1]; both need a machine-readable statement of what out-of-scope means. Layer 3 (Figure 2), the same allowlist diffed against egress flow logs, would let a lab attest isolation rather than assume it, which is the failure in the Anthropic name-collision case.

Limitations

- The positive fixtures are reconstructions of four to seven events from public descriptions; Hugging Face logged about 17,600 attacker actions [2]. No lab has released raw transcripts, so nothing here measures detection inside OpenAI, Anthropic or AISI, and "events earlier" describes our reconstruction.
- The fixtures and the rule share an author, so the specification tests show consistency, not generalisation. The DSEWiki run fixes the events but not the scope: its two non-wildcard allowlists are ours, and a different reconstruction would move the first page. The wildcard row, the scope the sandbox had, does not move.
- The live run is eight samples on tasks with no network, so it can only show silence. A positive live case needs an evaluation with internet on and a real allowlist. Wiki revision bodies are also a weaker kind of declared action than a curl argument: some URLs are links posted for other agents, not destinations reached.
- Sentinel reads declared arguments. METR reports that roughly 7% of the transcripts it evaluated in the Hugging Face investigation were successfully spoofed in places [13]; an agent can state one destination and reach another. Layer 3 is the answer, and it is not built.
- The extractor is regular expressions. Destinations hidden in shell variables, base64, DNS exfiltration or IP-as-integer forms are missed, and the miss rate is unmeasured; publishing the extractor publishes its gaps (Appendix A). T1 and T2 false-positive rates are likewise unmeasured.
- The wildcard scope is one misspecification of many. A scope that is correct today and stale tomorrow is the realistic case, and we have not modelled it.

Future work

A month would add three things. First, a positive live case: an agentic evaluation run with internet on and an honest allowlist (ExploitGym's own firewall list is the natural start), reporting T3 page and T1/T2 warn rates on real traces. Second, layer 3 against Cilium Hubble or Squid access logs, reporting the declared-versus-observed disagreement rate, which is the attestation number. Third, an adversarial pass on the extractor with an obfuscation corpus, so the miss rate is a number rather than a paragraph.

6. Conclusion

The public containment failures of 2026 were not failures of monitoring technology. They were cases where the scope of an evaluation existed in prose and in nobody's tooling. We show that writing the scope down as an allowlist and checking each stated destination against it produces a verdict a third party can recompute, and that the verdict is only as good as the allowlist: too tight and it pages on permitted traffic, absent and it is silent on the escape. The same trace shows the verdict arrives after the escape when the escape goes through a permitted egress, and a third party's record of a second incident reproduces both failure modes on events we did not write. Sentinel is small on purpose: a checkable layer under the monitors and proxies that already exist, and the argument that the artifact to publish per evaluation class is the pair (trace, allowlist).

Code and Data

Code repository: <https://github.com/crishN144/sentinel-evals> (Apache-2.0, matching ExploitGym). Fixtures, allowlists, the ablation script, the filled record of Appendix C and the tests are in the repository; `python -m pytest` reproduces every number here with no model and no API key. The DSEWiki export [5] is used as published, with its manifest hash stated in Section 4.4 and a download script in the README.

Author Contributions

C.N. conceived the project, wrote the scanner, fixtures and tests, ran the ablation and wrote the report. Claude (Anthropic) was used as a coding and drafting assistant under the author's direction; every number was recomputed by the author from the repository.

References

- [1] OpenAI, 2026. Hugging Face model evaluation security incident (21 July, revised twice) and The Hugging Face incident and the road ahead. openai.com/index/hugging-face-model-evaluation-security-incident; openai.com/index/hugging-face-incident-and-the-road-ahead
- [2] Hugging Face, 2026. Agent intrusion: technical timeline. huggingface.co/blog/agent-intrusion-technical-timeline
- [3] Anthropic, 2026. Investigating incidents in our cybersecurity evaluations. anthropic.com/news/investigating-incidents-cybersecurity-evals
- [4] UK AI Security Institute, 4 August 2026. Incident report: unsanctioned agent behaviour during cyber testing (INC-2026-07-28-01). aisi.gov.uk/blog/incident-report-unsanctioned-agent-behaviour-during-cyber-testing
- [5] Von Arx, S., Slade Byrd, C. and the Nightingale Collective, 4 September 2026. Discovery of a new OpenAI agent message board: DSEWiki edit dataset and report. collusion.wiki. OpenAI acknowledgement of 5 September 2026 as reported by TechCrunch, techcrunch.com/2026/09/05/openai-confirms-wiki-incident-says-its-working-on-a-framework-for-more-disclosure/
- [6] Meridian Labs and UK AI Security Institute, 2026. Inspect Scout: in-depth analysis of AI agent transcripts. github.com/meridianlabs-ai/inspect_scout
- [7] UK AI Security Institute and Redwood Research, 2025. ControlArena. github.com/UKGovernmentBEIS/control-arena

- [8] Kutasov, J. et al., 2025. SHADE-Arena: evaluating sabotage and monitoring in LLM agents. [arXiv:2506.15740](#)
- [9] Apollo Research, 2026. Watcher: security for coding agents (Watcher Live and Watcher Analyze). [watcher.apolloresearch.ai](#)
- [10] ExploitGym, 2026. ExploitGym: can AI agents turn security vulnerabilities into real attacks? [arXiv:2605.11086](#); repository and docs/firewall.md on GitHub, Apache-2.0.
- [11] UK AI Security Institute, July 2026. Cheating behaviour in frontier model evaluations. [aisi.gov.uk/blog/cheating-behaviour-in-frontier-model-evaluations](#)
- [12] Nagarkar, C., 2026. scorer-integrity: calibrating LLM judges against mechanical ground truth in Inspect evals. [github.com/crishN144/scorer-integrity](#)
- [13] METR and Redwood Research, 26 August 2026. Brief independent investigation of agents' behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident. [metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation](#)
- [14] Baker, B. et al., 2025. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. [arXiv:2503.11926](#)
- [15] Apart Research and CeSIA, 2026. AI Incident Response Sprint, Resources: Track 1 example projects, item 5 (Egress attestation spec v0.1). [apartresearch.com/sprints/ai-incident-response-sprint-2026-09-11-to-2026-09-13](#)
- [16] SecureLayer7, 2026. Hugging Face AI agent intrusion: technical anatomy. [blog.securelayer7.net/huggingface-ai-agent-intrusion-technical-anatomy](#)
- [17] Elastic Security Labs, 2026. Detecting the AI agent attack chain behind the Hugging Face breach. [elastic.co/security-labs/ai-agent-attack-detection-hugging-face-breach](#)
- [18] Cloud Security Alliance. AI Controls Matrix, control AIS-13 (AI sandboxing), and vulnerability-operations guidance on exploit-confirmation environments. [cloudsecurityalliance.org](#)
- [19] Coalition for Secure AI, 2026. AI Incident Response Framework v1.0 (Workstream 2). [coalitionforsecureai.org/wp-content/uploads/2026/03/AI-Incident-Response-1.pdf](#)

Appendix A. Limitations and Dual-Use Considerations

Limitations are listed in Section 5 and are not repeated. Dual use is the concern specific to publishing a detector. Sentinel's extractor is a small set of regular expressions; publishing it publishes exactly which command forms it does not parse, which is an evasion map for an agent that reads the code. We publish it anyway for three reasons. First, the load-bearing component is the allowlist, not the extractor; an agent that evades the regex still has to reach a destination, and layer 3 (observed flows) does not depend on parsing. Second, the Track 1 criterion is third-party verifiability, which is impossible with a secret rule. Third, the labs' own accounts show the agents already read the evaluation's code and papers to find the flag [1, 11]; hiding a detector from an agent that greps for it is not a control. The mitigation we recommend is the one we could not build in three days: pair the declared-action check with an observed-flow check, so that spoofing the argument does not spoof the verdict.

The fixtures contain no working exploit. Every command in the positive traces is a stated destination with a comment; the SSRF payload, the credentials and the internal hostnames from the incident are not reproduced. This respects the sprint rule against releasing installation recipes. The taxonomy in Appendix B could in principle guide an agent towards quieter behaviour; it is drawn entirely from the labs' published accounts and adds nothing they have not stated.

Appendix B. Taxonomy coverage

Behaviour (from the public accounts)	Seen in	Sentinel
1 Grader / evaluation probing	OpenAI [1], AISI [11]	T1 WARN
2 Sandbox / boundary testing	OpenAI, Hugging Face [2]	T2 WARN
3 Unauthorised egress vs declared scope	all four	T3 PAGE
4 Inter-agent message boards	OpenAI (Artifactory WebDAV), AISI (GitHub), DSEWiki [5]	not covered
5 Credential harvesting and reuse	OpenAI, Anthropic	not covered (env telemetry)
6 Out-of-scope target enumeration	Anthropic (~9,000 targets)	T3 if a host is stated
7 Persistence past a safe exit	OpenAI (198/898 unsolved tasks)	not covered (policy)
8 Transcript / monitor tampering	AISI, METR [13]	not covered; defeats T3
9 Goal adoption from peers	OpenAI	not covered

Table 3. The nine behaviours distilled from the four public accounts, and which ones an allowlist-relative scanner can see. Only item 3 is a mechanical page; items 1 and 2 warn; the rest need environment telemetry or multi-agent corpora and are out of scope for this artifact.

Appendix C. A filled containment record for one evaluation class

Track 1 asks whether a third party could verify compliance without access to the lab's network. This is what that verification would consume: an example record a lab could attach to an evaluation class, filled here for the ExploitGym class on our reconstruction of the Hugging Face trace. Every hash is computed from the files in the repository by `experiments/make_record.py`, and a test fails if the record and the files disagree. A lab would attach its own log in place of the reconstruction; nothing else in the record changes.

Field	Value
Evaluation class	exploitgym-cyber-eval (filled 2026-09-14)
Declared scope	allowlists/exploitgym.json sha256 f6484df479ecce81f61f4e5b3861cd01f25696e0fbc2f572de20b7903e65dc93 allowed_hosts: api.anthropic.com, api.openai.com, artifactory.internal; internet_expected: false
Trace	traces/hf_openai.json (7 events) sha256 2aaaf0f915fa346821c43987207ef48ee60c674af734ba9597fa5ab2a48fcdf7 RECONSTRUCTION from the public record (huggingface.co/blog/agent-intrusion-technical-timeline; openai.com/index/hugging-face-model-evaluation-security-incident). A lab would attach its own log.
First out-of-scope stated destination	event 3: cybergym-fn-7bf.modal.run
Verdict	PAGE (3 warn-only findings, never paged)
Who can recompute it, and how	sentinel score traces/hf_openai.json the two files above and this repository; no network, no model, no API key core.py sha256 1691ce806d1624de..., extract.py sha256 4f3e4cc85fa4a478...
What the record does not establish	• that the escape was prevented (page != prevent) • anything about traffic through allowlisted hosts (SSRF via a permitted proxy is invisible) • anything about destinations the agent reached but did not state (declared, not observed, action)

Table 4. The filled record, verbatim from `records/exploitgym-cyber-eval.json`. A second party recomputes the verdict from the two hashed files and the repository, offline. The trace here is a reconstruction; the record format does not depend on that.