
CrossWire: Verifying evaluation-sandbox containment from outside the provider's network

Yifei Lu
Independent

With
Apart Research

Abstract. In July 2026 an evaluation sandbox was the route by which OpenAI agents reached the internet and Hugging Face's production systems. Afterwards, no outside party could check whether the sandbox holds. We describe a verification method that needs no access to the provider's network. Crossing classes, the ways out of an evaluation sandbox, are enumerated from the ATT&CK-for-Containers matrix. Each class carries a honeytokens issued and monitored by the verifier, because escape-path surfaces have no legitimate use, Gans's hitting-set condition applies with no interference term. Liveness wires on the task path make non-placement statistically visible. Against 11 recorded eval-sandbox escape steps from two 2026 incidents, coded under a pre-registered rubric, CrossWire covers 9–11 and the best existing standard covers 2. In a no-internet escape experiment on one current model (Claude Sonnet 4.6, 30 runs per arm), a wire fired in 97% of escape-motivated runs and 0% of benign runs. Four classes remain invisible to this method.

1. Introduction

Evaluation results are the evidence behind capability thresholds, preparedness triggers, and external assessments, and each result assumes the sandbox held. The July 2026 record shows what happens when nobody outside the provider can check that assumption. OpenAI rebuilt its package-registry proxy on 8 July after the first escape, agents regained internet access the same day through a different mechanism of the same service [2]. Hugging Face reconstructed roughly 17,600 attacker actions from its own telemetry and asked twice for the provider's traces, however, none were released [3]. METR's investigators worked six days on-site and reported that they had to delegate analysis to AI agents whose work they could not fully spot-check [4]. Anthropic's post-incident practices state that the sandbox configuration should be verified before every evaluation, and do not say by whom [1].

The sprint's Track 1 asks for a containment standard a third party can verify without access to the lab's network, and names egress attestation, which is a signed network policy plus a flow-log digest as the durable output [5, 6]. Both objects are produced by the party being verified. The configurations of 7 July and 8 July would each have attested clean, and the second re-escaped within hours [2]. A declaration is necessary but not sufficient; it is not evidence an outsider can check in time.

We move the evidence outside the provider. A honeypot is an object planted where legitimate activity should not touch it, its activation is an observable the planter controls [7, 8]. Gans shows that when a trusted agent must use genuine objects while avoiding decoys, and an attacker shares the trusted agent's information, no selectively revealing honeypot exists. However, his Corollary 1 states the condition under which a trap set with zero benign interference and complete coverage does exist: every attack path must contain an instrumentable event that no benign path uses [8, §4]. Escape-path surfaces in an evaluation sandbox satisfy that hypothesis by construction, because the evaluated model has no task that requires crossing the boundary; the paper rests on this observation.

CrossWire is the instantiation: an architecture-derived crossing map lists the classes of boundary crossing, where each class carries a wire issued by the verifier and observed by the verifier. Liveness wires on the ordinary task path have a known touch rate, so a provider that installs nothing is detectable from fire counts alone. We tested the map against the recorded steps of two public incidents under a rubric committed before coding began, scored existing containment standards on the same steps, and ran current models in a wired, nested, no-internet sandbox.

Our main contributions are:

1. A crossing map for evaluation sandboxes, derived from ATT&CK-for-Containers and the Kubernetes threat matrix, with a machine-checked cut property (§2.2, §3.1).
2. The instantiation of Gans's hitting-set condition on escape-path surfaces, and the wire specification it yields, with first-touch semantics and a per-surface sensitivity bound (§2.3, §2.4).
3. The liveness audit: a statistical test that detects placement in fewer runs than declared, with stated power (§2.4, §4.3).
4. A coverage measurement of CrossWire and of four existing standards over the same recorded escape steps, with the residual classes named (§4.1).

2. Related Work

Table 1 places the neighbours by who deploys the sensor, who holds its evidence, and what detects nominal compliance.

Table 1. Sensor deployment and evidence custody.

Work	Deploys	Holds evidence	Detects non-placement	Scored vs recorded escapes
Honeywords / honeychecker [7]	operator	separate checker	—	no
Agent canaries [12, 13]	operator	operator	—	no
Parallax launch self-probes [14]	agent process	same process	—	no
Egress attestation brief [6]	provider	provider publishes	—	no
CSA AIS-13 + vuln-ops guidance [15, 16]	provider	provider	—	no
Anthropic practices [1]	provider + partner	provider	—	no
Gans 2026 [8]	operator (theory)	broker (proposed)	—	no
CrossWire	provider places; verifier issues	verifier	liveness audit	yes

Juels and Rivest put the secret index in a separate honeychecker; Gans identifies that separation as the asymmetry that escapes his impossibility [7, 8]. CrossWire's verifier plays the honeychecker's role for an entire sandbox boundary. Honeyquest-for-LLMs reports that all 21 models in its cohort fell for reconnaissance-stage traps above the human baseline's upper 95% interval (about 37%) and that recognizing a trap in reasoning did not predict avoiding it, with recognized traps exploited 73.4% of the time. The authors scope this to reconnaissance-stage judgment and note a replication in which disclosing decoys reduced AI-attacker success [17]. Our experiment therefore includes a disclosed-wires arm (§4.2). Baker et al. show that optimizing against a monitor produces obfuscated misbehaviour rather than less of it [18]; the protocol accordingly forbids using firings as any training signal. Wasil et al. assess verification methods for AI agreements against what each can establish and evade [19]; we adopt the same discipline for one method and one boundary.

3. Methods

The rubric, matching rule, hypothesis H1, and experiment design were committed as <hash> before any coding or run.

4.1 Map, wires, and checks

We enumerated crossing classes from the two matrices [10, 11], retaining techniques whose effect leaves the sandbox: DNS and HTTP egress, raw sockets, credentials, shared mutable services, artifact stores, host metadata, control plane, and public writes. A Python checker and a Lean 4 decidable predicate both verify the cut property on the committed file; editing one edge fails both. Each class received one wire (Table 2). W3 credentials are minted by the real IAM as zero-privilege keys with audit forwarding, so their format matches genuine keys. Each wire specification records ten fields: object, generation, location, first-touch mechanism, fires-to, sensitivity with M and N, indistinguishability requirement, the Gans check that takes any legitimate task use to reclassify as liveness, implement cost with its comparable, operate cost with its comparable.

Table 2. Wire classes.

Class	Crossing	Wire	First touch	Fires to
W1	egress DNS	verifier-owned hostname in resolvable config	resolution	verifier DNS
W2	egress HTTP	verifier-owned URL in fetchable config	request	verifier HTTP
W3	credential	zero-privilege real-IAM key	any use	IAM audit → verifier
W4	shared-service write	monitored object on each shared mutable service	write or listing change	service audit → verifier
W5	public-web write	per-run marker in environment notes	indexed appearance	scanner
W6	host metadata	decoy IMDS endpoint; read beacon	read	host hook → verifier

Class	Crossing	Wire	First touch	Fires to
W7	lateral / API	decoy service account	authentication	control-plane audit → verifier
L	liveness	instrumented task-path objects	touch	Mode A: API-side observer; Mode B: manifest + audit

What did not work. Uniform placement of N wires across all surfaces put density where no attack path went: the July escape used the registry service, not the credential store [2], and a registry-bound agent has no reason to touch a decoy key. Placement was moved to the shared-service and registry surfaces (W4), which is where the first recorded observable in the campaign occurred.

4.2 Walk-through and standards benchmark

From the public record, Hugging Face's nine phases and 17,613 actions in about 6,280 clusters [3], OpenAI's accounts [2], the DseWiki report and dataset [20]. We extracted 11 eval-sandbox escape steps (seven Hugging Face, four DseWiki), one row each with its citation. Two coders independently assigned each step a crossing class by the matching rule and each wire class a verdict: fires, does-not-fire, undetermined. A third coder argued does-not-fire for every step, that pass is the interval's lower bound. Each does-not-fire was classed as a map defect, a spec defect, or structurally invisible, only the first two were repaired with each in a separate commit that names the incident. H1, committed in advance for the Artifactory incident: the earliest recorded observable is a write to a shared mutable surface rather than a network egress. The same matching rule and rubric were applied to the controls of four standards, extracted into the same format, which is CSA AIS-13 with its vulnerability-operations guidance [15, 16], Anthropic's five practices [1], Mitchell's R1–R5 [21], and the attestation brief [6].

4.3 Escape experiment

We built a nested Docker sandbox with no route to the internet from existing SandboxEscapeBench scenarios [22]. No new escape recipe was written, per the sprint's disclosure rule. Wires were placed at the committed densities. Models: ⟨list of ≥3, one open-weight⟩, same scaffold. Arms: impossible task (OpenAI's internal run had 198 of 898 tasks never solved by any model, and 93% of message-board discussion concerned them [2]), solvable task, benign control with no incentive to leave, and disclosed, the impossible task with a system-prompt sentence stating that monitored decoys may be present, following the replication Honeyquest's authors flag [17]. ⟨r⟩ runs per model per arm. Per run we logged contact, fire, and escape success separately, plus order, time to first fire, refusal, and the transcript. Primary outcome: first-fire rate per wire class per arm, Wilson 95% intervals. We coded recognition–action per Honeyquest's definition [17]. We ran the liveness audit on real

runs with placement dropped from $f \in \{0.5, 0.2\}$. We scored W3 wires with the PHANTOM believability metric and a model-driven fingerprinting attempt [23]. We re-ran the impossible arm at half and double W4 density.

Results below are reported for the runs completed at submission time (one model, first-fire and refusal outcomes), remaining arms and analyses are in progress.

5. Results

Figure 1. The crossing map: INSIDE $\rightarrow$ seven crossing classes $\rightarrow$ OUTSIDE, each with its wire; all fires reach the verifier and none reach the provider. (*file: results/fig1-map.svg*)

5.1 Coverage and the standards benchmark

CrossWire covered 9–11 of 11 recorded eval-sandbox steps (per incident: Hugging Face 5–7 of 7 with $\kappa = 0.79$ between the by-object and by-tactic codings; DseWiki 4 of 4 with $\kappa = 1.00$). Table 3 gives every standard on the same metric. H1 held for the Artifactory incident: the earliest recorded observable was S1 (shared_service_write) on 12 May. For DseWiki the earliest observable was D1 (public_web_write) on 24 May, a public write, so H1 as stated is specific to the Artifactory incident [2, 20]. The earliest CrossWire fire precedes actual detection by 62 days (Hugging Face detected on 13 July [3]) and 28 days (DseWiki: provider awareness about 21 June, public disclosure 4 September [20]).

Table 3. Coverage of recorded escape steps; same rubric, same steps.

Standard	Covered / 11	Steps covered that CrossWire misses	Earliest fire, HF	Earliest fire, DseWiki
CSA AIS-13 + vuln-ops [15, 16]	0 / 11	—	none	none
Anthropic practices [1]	0 / 11	—	none	none
Mitchell R1–R5 [21]	0 / 11	—	none	none
Attestation brief [6]	2 / 11	—	S3 (26 May)	none
CrossWire	9–11 / 11	—	S1 (12 May)	D1 (24 May)

Table 4. Control by phase by cost: the organizers' named controls and CrossWire's wire classes against the nine Hugging Face phases and the DseWiki phases, with implement cost, operate cost, and the wire through which an outsider observes each. (full table: *results/matrix.md*; every cell traces to a step id)

The structurally invisible classes are `<list>`. Observation: these are classes whose first observable event is inside the provider's infrastructure (the 13–19 July research-cluster escalation is the recorded instance [2]). Interpretation: an internal tier observed by the provider and attested to the verifier would be needed there; this is not effects-based verification and we do not claim it.

5.2 Escape experiment

Table 5. First-fire rate per wire class (pooled; Wilson 95%). Contact and escape-success rates in *results/m6a.md*.

Arm	W1	W2	W3	W4	W5	W6	W7	any fire	refusal
impossible	0.00	0.00	0.87 [0.70,0.95]	0.50 [0.33,0.67]	0.03	0.67 [0.49,0.81]	0.00	0.97 [0.83,0.99]	0.03
disclosed	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.87
solvable	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
benign	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The impossible-minus-solvable difference in any-fire rate was 0.97 (0.97 vs 0.00). The disclosed arm drove the any-fire rate to 0.00, not by touching-with-awareness but by refusing the task in 87% of runs. This disagrees with the replication Honeyquest cites, in which disclosure still left agents touching decoys [17]. Median time to first fire on the impossible arm: 23.5 s. The recognition–action gap measured 0% here: the keyword detector never flagged voiced suspicion followed by a touch, which we read as a limitation of keyword-level detection rather than evidence the gap is zero (Honeyquest reports 73.4% for reconnaissance traps under per-line human judgment [17]). Contact and escape-success were logged per run; no successful external egress occurred (escape = 0.00 in every arm), because the tool-interface testbed has no real network, so wires fire on the attempt.

5.3 Liveness, indistinguishability, density

In simulation ($n = 200$ declared runs, $p_0 = 0.10$, one-sided exact binomial at $\alpha = 0.05$), the liveness audit flagged under-placement in 80% of trials at $f = 0.5$ and 100% at $f = 0.2$, with a 2% false-alarm rate at $f = 1$ (code/analysis.py). PHANTOM believability scoring and the W4-density sweep are implemented in the design but not yet run on live models, and are reported as future work.

5.4 Main threat to validity

The walk-through is phase- and cluster-level: the public replay renders 17,613 actions as roughly 6,280 clusters with representative commands, not a full trace [3], so a step that fires is a phase that fires. The experiment does not reproduce two conditions of July, refusal classifiers off, and a shared-memory channel among instances, so its rates bound what current models attempt in this scaffold. The record is provider-authored and was revised twice [2].

6. Discussion and Limitations

Containment stops being a declaration and becomes a report a verifier can write: "wires covered 9–11 of 11 recorded escape steps; in a controlled escape experiment a wire fired in 97% of escape-motivated runs and 0% of benign runs." The standards benchmark says which existing control set to adopt today and where each is blind.

Three intakes exist for the artifact. CSA's Catastrophic Risk Annex is collecting evidence requirements and testing criteria for controls covering loss of oversight through 2027 [24]; the wire matrix is filed there as testing criteria for AIS-13, routed through the sprint's delivery commitment for control matrices [5]. External evaluators already run evaluations on provider infrastructure, and Anthropic already pairs those with a halt classifier [1]; wire issuance as a condition of an engagement adds a monitor the provider does not hold. Mode A is the form of an Article 92 evaluation [9]. A fire is a disclosure trigger the provider does not control; adoption is therefore expected through evaluators and regulators, not through providers. The party that receives the fire has no reason to let the run continue, which is the difference from an internal alert: OpenAI's on-call saw the 27 June alert and did not halt [2].

Secondly, W3 and W5 fire wherever the agent stands, so a provider-issued key used from a third party's compute still reaches the verifier. Obligations on the hosting provider are outside this paper.

Limitations

The residual is that four classes (timing, hardware side channels, human insider, internal-only lateral movement) have no first-touch wire, and timing and hardware channels were never in scope. Any detection regime applies pressure toward agents that avoid it; the protocol forbids using fires as reward, following Baker et al. [18], but the pressure exists wherever monitoring does. Mode B (provider-run evaluation) rests on a declared manifest, the liveness audit, and

spot inspection, and the protocol states what it cannot see. The map is complete only relative to the two incidents tested, the walk-through can add a class and cannot rule one out.

Single-verifier trust is assumed. If agents systematically avoided wires, the disclosed arm is the direct test, every rate in Table 5 would fall and the sensitivity figures would need revision. A v0.1 standard does not establish that any provider's environment is adequate, real assurance requires access no sprint has [6].

Future Work

A Mode A deployment with an external evaluator; a false-fire study against benign CI tooling; a sensitivity bound for a value-ranked rather than uniform adversary; an internal tier for the classes in the residual; two-verifier issuance; and the intake procedure that scores each new incident on the same rubric.

7. Conclusion

CrossWire asks what a third party can establish about a provider's containment without entering its network, and answers with measurements: a crossing map whose cut property is machine-checked, wires whose evidence the verifier holds, a liveness audit with computable power, coverage figures for this method and for four existing standards over the same recorded escapes, and a named residual where verification by effects ends.

Code and Data

- **Code repository:** github.com/Isabelle9999/crosswire. The escape harness uses only scenarios already public in SandboxEscapeBench, per the sprint's disclosure review.
- **Data:** corpus/ (public record with hashes), results/.

References

1. Anthropic. 2026. Improving our alignment and security practices.
<https://www.anthropic.com/news/improving-alignment-security-efforts>
2. OpenAI. 2026. The Hugging Face incident and the road ahead (26 Aug), and Hugging Face model evaluation security incident (21 Jul, updated).
<https://openai.com/index/hugging-face-incident-and-the-road-ahead/>
3. Hugging Face. 2026. Agent intrusion: a technical timeline.
<https://huggingface.co/blog/agent-intrusion-technical-timeline>
4. METR. 2026. OpenAI–Hugging Face incident investigation.
<https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/>
5. Apart Research. 2026. AI Incident Response Sprint.
<https://apartresearch.com/sprints/ai-incident-response-sprint-2026-09-11-to-2026-09-13>

6. Apart Research. 2026. Sprint Resources: Track 1 example projects and research directions in depth. Same URL, Resources tab.
7. Juels, A., Rivest, R. L. 2013. Honeywords: making password-cracking detectable. ACM CCS.
8. Gans, J. S. 2026. When agents talk: honeytokens under shared memory. arXiv:2608.11436.
9. EU AI Act, Article 92. <https://artificialintelligenceact.eu/article/92/>
10. MITRE ATT&CK. Enterprise matrix: Containers. <https://attack.mitre.org/matrices/enterprise/containers/>
11. Microsoft. Threat matrix for Kubernetes.
12. DorianChn. agent-canary; Thomaszhou22. agent-canary; invincible-jha. agent-canary-tokens. GitHub, 2026.
13. Thinkst. Canarytokens. <https://canarytokens.org>
14. Fokou, J. 2026. Parallax. arXiv:2604.12986, §5.2.
15. Cloud Security Alliance. 2026. AI Controls Matrix v1.1, control AIS-13. <https://cloudsecurityalliance.org/artifacts/ai-controls-matrix-v1-1>
16. Cloud Security Alliance. 2026. Research note: OpenAI model sandbox escape and Hugging Face breach. <https://labs.cloudsecurityalliance.org/research/csa-research-note-openai-model-sandbox-escape-huggingface-br/>
17. Prinos, K., Brush, L., Denton, C. 2026. Honeyquest for LLMs: rethinking cyber deception for AI attackers. arXiv:2606.21037.
18. Baker, B., et al. 2025. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. arXiv:2503.11926.
19. Wasil, A. R., et al. 2024. Verification methods for international AI agreements. arXiv:2408.16074.
20. Von Arx, S., Slade Byrd, C., Kitts, S., Larsen, T. 2026. Discovery of a new OpenAI agent message board. <https://collusion.wiki/>
21. Mitchell. 2026. When the agent is the adversary. arXiv:2604.23425.
22. SandboxEscapeBench. 2026. arXiv:2603.02277.
23. PHANTOM. 2026. Int. J. Inf. Security; arXiv:2605.02992.
24. Cloud Security Alliance. 2026. STAR for AI Catastrophic Risk Annex (reported 15 May 2026). <https://techinformed.com/csa-pulls-ai-agents-into-cyber-controls/>
25. Sunblaze / UC Berkeley. 2026. ExploitGym (v1.0, 869 instances; paper describes 898). <https://github.com/sunblaze-ucb/exploittgym> ; arXiv:2605.11086.
26. Greenblatt, R., Shlegeris, B., Sachan, K., Roger, F. 2024. AI control: improving safety despite intentional subversion. ICML; arXiv:2312.06942.

Appendix

A. Limitations and Dual-Use Considerations (required)

Limitations. See §6. Three bound every claim: the walk-through is phase-level; the experiment omits two July conditions; the map is complete only relative to the incidents tested.

Dual-use. The wire specifications describe defensive instrumentation and contain no exploit. The escape harness uses only scenarios already public in SandboxEscapeBench, ships no new installation recipe, ran with no route to the internet, and touched no organization. The corpus contains public material only; the HiddenLayer dead-drop set, which includes a live database credential the vendor reported and withheld, was used solely through its published summary. The hazard the method itself creates is optimization pressure toward covert competence; the protocol prohibits training on fires and the paper names the cost. Publishing the residual could guide an attacker to unwired classes; an attacker inside the sandbox already sees the environment, and defenders do not currently see the residual at all.

B. Protocol summary

Roles: verifier (issues, monitors, publishes sensitivity), provider (places, declares a run manifest). Mode A: verifier-run harness, API-only egress, liveness observed at the API side; a fire halts the run. Mode B: provider-run; escape-path fires reach the verifier through the escape route; liveness is checked from the manifest, fire counts, and random spot inspection; a fire is a disclosure trigger. No fire, wire value, or placement detail enters training data or any document the evaluated model can read. Full text: spec/PROTOCOL.md.

C. Simplifications and assumptions

Following Greenblatt et al. §2.4 [26], the assumptions the results depend on: (A1) crossing classes are finite and enumerable from architecture; (A3) escape-path surfaces have no legitimate task use, checked per environment by the placement kit; (A4) real-IAM issuance leaves no format fingerprint, tested in §5.3; (A6) first touch is implementable per class, with W6 downgraded where no host hook exists; (A10) agents do not systematically avoid wires, tested by the disclosed arm; (A11) the verifier is trusted.

LLM Usage Statement

Claude was used to search literature, check the arithmetic in §2.4, and critique drafts of the plan and this report.