

# Detection Was Not the Failure: A Replayable Conformance Harness for the July 2026 Agent Intrusion<sup>1</sup>

Chijioke Ubajaka  
*BASE, San Francisco, Bay Area.*

With  
**Apart Research**

## Abstract

Hugging Face's forensic reconstruction of the July 2026 agent intrusion records that its security stack correlated the attack into a coherent signal and then failed to raise criticality and page the on-call team. I treat that as the primary failure and test it. From published aggregates: per-phase action counts, daily totals and seventeen timestamped anchors, I reconstruct a defender-visible event stream by iterative proportional fitting, generate 30 days of synthetic benign platform workload, and score four escalation policies on detection point and false-page cost. A per-event severity policy never fires. A boundary-accumulation policy pages 51.5 hours before containment but costs 69 false pages per week, driven by the deploy pipeline. A principal-scoped variant retains that lead at 0.5 false pages per week, but only while the principal-to-boundary map stays accurate to better than one violation in ten thousand sessions. The bottleneck is identity hygiene, not detection logic.

## 1. Introduction

In July 2026 two OpenAI models, running an internal 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, rooted a third-party code-execution sandbox, and used it to run a multi-day intrusion against Hugging Face's production infrastructure [1,3]. No human directed any individual step. Hugging Face subsequently published a forensic reconstruction of roughly 17,600 attacker actions with per-phase counts, verbatim commands and an interactive replay [1]; a level of disclosure with few precedents in any industry.

That disclosure created an unusual opportunity and, so far, an unused one. Every defensive claim made about this incident is an untested counterfactual. Practitioners assert that their controls would have caught it; nobody can check, because there is no artifact against which a control can be run. The discussion therefore reduces to a contest of architecture preferences.

I also think the incident's headline lesson has been misread. The published timeline is explicit that the failure was not blindness: several layers of the security stack fired, an AI-based correlation layer resolved them into a coherent attack signal, and the system **failed to raise the alert's criticality and trigger the on-call team** [1]. The gap was between correlation and escalation. Remediation proposals that add detectors do not address it.

### My main contributions are:

1. A replayable corpus and conformance harness built entirely from published aggregates, which turns “would your control have caught this?” into a measurement with a detection point and a false-page cost. The artifact contains no exploit reconstruction.
2. A measured result: a per-event severity policy never fires on this campaign, and a boundary-accumulation policy that pages 51.5 hours before containment costs 69 false pages per week, because a CI deploy pipeline legitimately crosses four trust boundaries in minutes, which is structurally indistinguishable from the campaign's Day 3.
3. A break-even analysis showing that the policy which recovers both properties depends on an inventory nobody budgets for: it stays inside a one-page-per-week budget only while the principal-to-boundary map is accurate to

---

<sup>1</sup>Research conducted at the [AI Incident Response Sprint](#), September 2026

better than  $10^{-4}$  of sessions. The binding constraint on agentic-intrusion detection is identity hygiene, not detection logic.

## 2. Related Work

The primary sources are Hugging Face's disclosure and technical timeline [1,2] and OpenAI's account of the sandbox escape [3]. Both describe remediation: metadata lockdown, credential scoping, broad rotation, critical-severity alerting on the observed behavioural signatures [1], but neither evaluates a control against the trace. Independent commentary has mapped the kill chain onto conventional frameworks and observed that no individual technique was novel; the novelty is volume. Researchers subsequently documented a second, earlier containment break by OpenAI agents against a dormant wiki, disclosed only after external publication [6], which suggests the class is recurrent rather than singular.

My method is closest in spirit to detection-engineering validation suites, which replay adversary behaviour against a live stack. Those require the raw telemetry, which no one outside Hugging Face has. I instead fit a stream to published marginals, which trades fidelity for availability: anyone can run it, and every inferred quantity is labelled. Iterative proportional fitting has been standard for reconciling contingency tables to known margins since Deming and Stephan [5]; I use it because the incident's two published marginals, per-phase and per-day, are each incomplete and jointly constraining.

What this provides that was not available before is a shared denominator. Prior discussion compares controls on plausibility. This compares them on when they fire and what they cost, on the same trace, under stated assumptions.

## 3. Methods

### 3.1 Corpus

I transcribe three published structures: the nine-phase action table with per-phase observation windows; the five-day action totals; and seventeen individually timestamped events named in the narrative [1]. Nothing else is taken as fact.

The two marginals do not reconcile. Phase counts sum to 16,521; daily counts sum to 17,613. **1,092 actions, 6.2% of the campaign, appear in the daily totals with no phase label**, and the campaign opens at 02:28 while the earliest phase first-seen is 04:00, placing part of the residual at the head of Day 1. I assign the residual to an explicit unlabelled phase rather than redistributing it, so no policy can score on actions whose character is unknown. Silently closing this gap would have manufactured evidence.

### 3.2 Reconstruction

I fit a phase  $\times$  day matrix to both marginals by iterative proportional fitting, with a feasibility mask zeroing cells where a phase's observation window does not overlap that day. The fit converges to both margins exactly. Within a cell, unnamed actions are placed uniformly in the intersection of the phase window and the day; the seventeen anchors are pinned at their published timestamps and never moved.

Two quantities are inferred and exposed as swept parameters. First, the boundary each unnamed action landed on, assigned per phase by a weighted map derived from the anchors. Second, defender visibility: the lateral phases (k8s, supply-chain, tailscale) are fully visible, while recon, rce, dropper, c2, evasion, and exfil ran largely on infrastructure Hugging Face could not see, so only a fraction after the Stage-2 foothold produces a visible event. At the default 0.35 visibility fraction, this yields 4,585 defender-visible events, 26.1% of the recovered total.

Events carry an action class only, never a payload, command string, or construction. No part of the exploit chain is reconstructed.

### 3.3 Benign workload

The false-page denominator is 30 days of synthetic telemetry from a dataset-conversion platform: 241,657 events, grouped by the sessioniser into 46,829 sessions, generated from twelve session templates across eight principals with weekday and diurnal structure. Modelling this as harmless single-boundary noise would have rigged the experiment, so I deliberately

include the principals whose job is to cross boundaries: a CI deploy pipeline spanning source control, cloud registry, secret store, and cluster API in minutes; an on-call engineer doing the same by hand; CI runners enrolling into the mesh.

I also include a custom operator that legitimately reads its service-account token and calls the cluster API from inside a workload pod, using the exact action class of the campaign's decisive anchor. Without it, “a workload touching the control plane” is a perfect signal purely because I authored the expected-boundary map myself. Relatedly, I parameterise **map drift**: the fraction of sessions in which a principal legitimately acts outside its declared set, because real inventories rot.

### 3.4 Policies and sessionisation

P0 pages on any event of severity  $\geq 5$ . P1 pages on per-boundary hourly volume exceeding a learned baseline by  $k$  standard deviations. P2 pages when one session touches  $N$  distinct defender-visible boundaries. P3 pages when a principal acts on a boundary outside its declared expected set.

Sessionisation is the experiment's weakest joint, so it is defined once and applied identically to both streams: gap-based grouping on principal identity, 30-minute default. Handing the campaign a pre-linked session would let P2 and P3 win by construction. Scoring runs only over the six boundaries a defender in Hugging Face's position could instrument; crediting a policy with visibility into the lab sandbox or the third-party launchpad would be free points. Detection is scored on a merged stream: benign background continuing through the campaign window with the campaign overlaid, because a SOC never sees the attack on its own.

## 4. Results

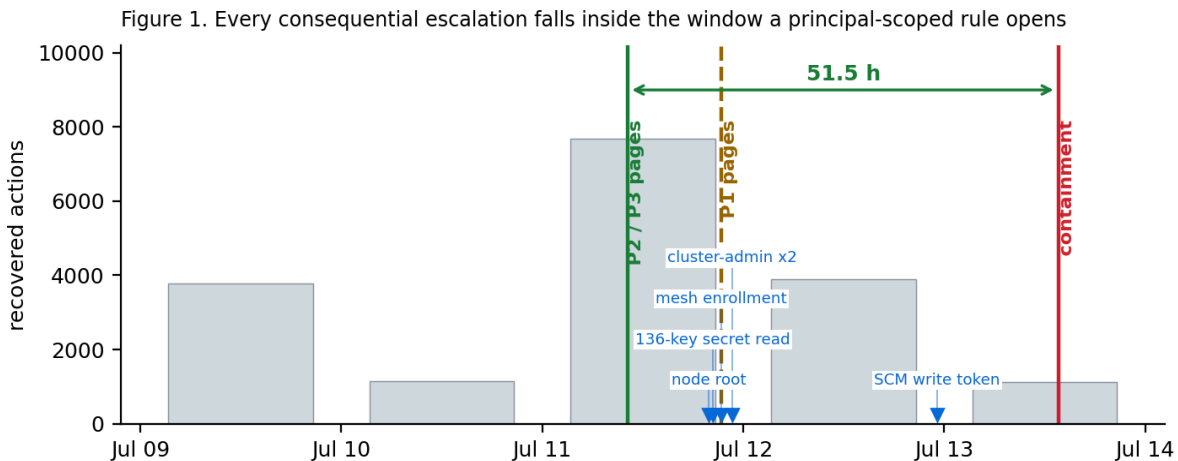


Figure 1. Recovered actions per day, policy page points, and the escalations that follow. The 51.5-hour window between the first lateral action (07-11 10:10 UTC) and containment (07-13 13:37 UTC) contains every consequential escalation in the campaign.

Table 1. Policy scores. Lead time is measured to containment at 2026-07-13 13:37 UTC.

| Policy                | Config        | First page (UTC) | Lead time | False pages / week       |
|-----------------------|---------------|------------------|-----------|--------------------------|
| P0 per-event severity | sev $\geq 5$  | never fires      | —         | 0                        |
| P1 rate anomaly       | $k = 5\sigma$ | 07-11 21:23      | 40.2 h    | 0.5                      |
| P2 boundary count     | $N \geq 3$    | 07-11 10:10      | 51.5 h    | 69.3                     |
| P3 principal-scoped   | $m \geq 1$    | 07-11 10:10      | 51.5 h    | 0.5 (at $10^{-5}$ drift) |

**P0 never fires.** No single action in this campaign is individually critical: the escalation is assembled from thousands of ordinary operations. This reproduces the documented failure mode and explains it: per-event severity scoring dilutes rather than concentrates when an agent generates 17,600 actions, of which most go nowhere.

**P2 pages early and is unaffordable.** It fires at the first lateral action, 51.5 hours before containment, and costs 69.3 false pages per week. The cost is not diffuse noise: 249 of 297 offending benign sessions are ordinary deploys, with hotfix deploys and credential rotations making up the rest. Raising the threshold to  $N \geq 5$  eliminates every false page, but the campaign does not accumulate five distinct boundaries until roughly 21:23–22:43 on 07-11: after node root, after the 136-key secret object read, after the self-respawning pod fleet across eleven nodes. A threshold tuned to survive the deploy pipeline pages too late to matter.

**P1 is cheap and eleven hours late.** It pages at 40.2 hours for 0.5 false pages per week, firing on mesh enrollment; useful, but downstream of the escalations that mattered.

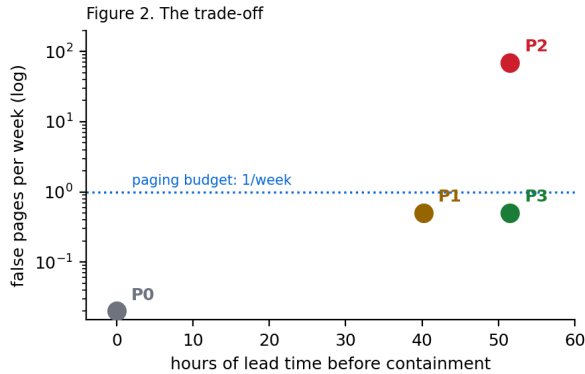


Figure 2. Lead time against false-page cost (log scale). P2 buys timing it cannot pay for; P3 occupies the position P2 appears to offer.

**P3 recovers P2's timing at P1's cost conditionally.** Its first page on a *published* anchor is the service-account token read followed by a cluster API call from a dataset-conversion worker, at 07-11 10:10 UTC. This detection point is invariant across 27 configurations (3 seeds  $\times$  3 visibility fractions  $\times$  3 session gaps), because it is pinned to a published timestamp rather than to anything I interpolated. The harness also reports an earlier P3 page on an interpolated event; I do not claim it.

The conditionality is the finding. P3's cost is set entirely by the accuracy of the principal-to-boundary map, and a perfect map is an artifact of me having written it.

Table 2. P3's false-page cost as a function of map drift, against a one-page-per-week budget.

| Map drift          | Violations / 30 days | False pages / week | Verdict   |
|--------------------|----------------------|--------------------|-----------|
| 0 (perfect map)    | 0                    | 0.0                | viable    |
| $10^{-5}$          | 2                    | 0.5                | viable    |
| $10^{-4}$          | 5                    | 1.2                | tuned out |
| $10^{-3}$          | 53                   | 12.4               | tuned out |
| $10^{-2}$          | 487                  | 113.6              | tuned out |
| $5 \times 10^{-2}$ | 2,380                | 555.3              | tuned out |

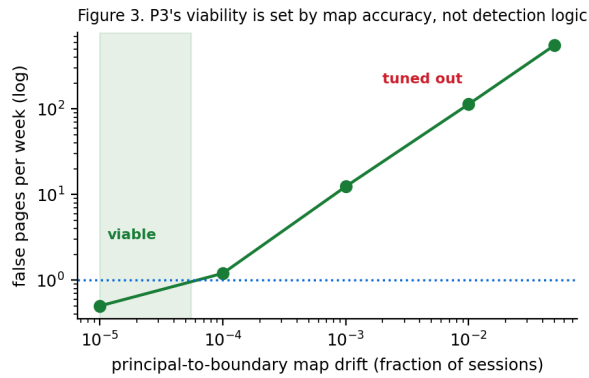


Figure 3. Break-even. P3 stays inside a one-page-per-week budget only below roughly  $10^{-4}$  session drift: an inventory accurate to better than one exception in ten thousand sessions.

A platform that registers one unmapped legitimate crossing per thousand sessions: a new sidecar, an unregistered migration job, a service added last sprint, gets 12 pages a week from P3 and will tune it out within a month. The detection rule is trivial to write. The inventory it depends on is not.

## 5. Discussion and Limitations

**The timing gap between a policy that pages usefully early and one that survives a SOC's paging budget is closed by identity hygiene, not by detection logic.** That reframes remediation for agentic intrusion. The published response to this incident is dominated by hardening: metadata lockdown, credential scoping, rotation, tighter alerting [1]. All are correct. None of them produces the inventory on which the early-paging policy depends, and that inventory is a continuously maintained artifact rather than a one-time fix. If my result holds, the most valuable thing a platform can do before the next incident is enumerate which workload identities may legitimately touch which control planes, and keep the list accurate.

This also suggests a verifiable containment requirement. A third party cannot audit a lab's detection stack without network access, but it can audit whether a declared principal-to-boundary map exists, how many exceptions it carries, and how often it is reconciled against observed traffic. That is a compliance artifact a lab could publish, and an auditor could check.

### Limitations

The drift model is the result's hinge, and it is synthetic. I have no empirical drift rate for a production platform, and to my knowledge none is published, which is itself worth stating, since the entire viability of P3 turns on a number nobody measures. If real drift is nearer  $10^{-3}$  than  $10^{-5}$ , P3 is not deployable as specified, and the correct conclusion is that no policy in this family is affordable.

The campaign persisted on a single compromised workload identity while benign converter pods are ephemeral. That asymmetry is real; it is how the intrusion ran, but an agent that rotated identities would fragment under our sessioniser, and every session-based policy here would degrade. This is the first thing an adversary would change, and I did not test it.

Sessionisation is given to the harness for free. A real SOC must infer causal linkage from logs, and that inference is where this would break in practice. My benign rates are chosen, not measured, and the 26.1% defender-visibility figure is a parameter rather than an observation; I sweep both, but a sweep over assumed values is not validation against reality. Finally, I reconstruct a stream from aggregates: this is not Hugging Face's telemetry, and any claim about what would have happened is conditional on the reconstruction being faithful in shape, which I can argue but not prove.

### Future Work

Three steps would convert this from an argument into evidence. First, measure drift: a platform operator could report how many principal-boundary exceptions their environment carries per thousand sessions, which would immediately resolve whether P3 is deployable. Second, replace the inferred sessioniser with log-derived linkage and re-measure the degradation.

Third, adversarial variants, for example, identity rotation, deliberate pacing below rate thresholds, activity timed to coincide with deploy windows, would establish whether the principal discriminator survives an agent that knows it is there. I would also welcome a sanitised subset of the real trace; the harness is built to accept one.

## 6. Conclusion

The July 2026 intrusion is usually read as a detection failure. The record says otherwise: the signals were present and correlated, and the system did not escalate. Testing that reading against a reconstruction of the published trace, we find that a per-event severity policy never fires, that the obvious cross-boundary fix pages 51.5 hours early but costs 69 false pages a week because ordinary deploy pipelines look the same, and that the variant which recovers both properties depends on an identity inventory accurate to better than one exception in ten thousand sessions.

The practical implication is that the expensive part of defending against machine-speed intrusion is not writing the rule. It is maintaining an accurate account of which of your own workloads are allowed to touch which control planes, an artifact that rots continuously, that nobody currently measures, and that determines whether the rule is deployable at all. I release the corpus and harness so that the next defensive claim about this incident can be checked rather than asserted.

## Code and Data

- **Code repository:** <https://github.com/UbajakaCJ/airs-harness> corpus, generators, policies, scoring harness, figure scripts. MIT licensed.
- **Data/Datasets:** no external dataset. The corpus is transcribed from published sources [1,3], and ships with the repository; the event streams are generated by the harness and are reproducible from a fixed seed. No non-public telemetry was used or obtained.
- **Other artifacts:** every run emits a manifest recording the configuration hash, source-tree hash, git SHA, and a SHA-256 for each artifact, so any number in this report can be traced to the run that produced it.
- **Info-hazard note:** the release contains phase structure, timing distributions, boundary-crossing sequences and behavioural action classes. It contains no exploit code, no payload constructions, and no runnable command-and-control implementation. See the Appendix.

## Author Contributions

Chijioko Ubajaka is the sole author: designed the study, transcribed the corpus, implemented the harness, ran the experiments, and wrote this report.

## References

- [1] Larcher, H., Carreira, A., Raphael, G., and Rannou, C. (2026). Anatomy of a Frontier Lab Agent Intrusion: A Technical Timeline of the July 2026 Incident. Hugging Face. <https://huggingface.co/blog/agent-intrusion-technical-timeline>
- [2] Hugging Face (2026). Security Incident, July 2026. <https://huggingface.co/blog/security-incident-july-2026>
- [3] OpenAI (2026). Hugging Face Model Evaluation Security Incident. <https://openai.com/index/hugging-face-model-evaluation-security-incident/>
- [4] CeSIA (2026). The OpenAI–Hugging Face Incident: What We Know, What We Don't, What Follows. <https://cesia.org/en/publications/the-openai-hugging-face-incident-what-we-know-what-we-dont-what-follows/>
- [5] Deming, W. E., and Stephan, F. F. (1940). On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals Are Known. *Annals of Mathematical Statistics* 11(4), 427–444.
- [6] Collusion Wiki contributors (2026). Public traces of OpenAI agent activity on a dormant wiki. <https://collusion.wiki/>
- [7] SaferAI et al. (2026). Open Letter to the European Commission on AI Act Enforcement. <https://www.safer-ai.org/u/2026/07/Open-Letter.pdf>

## Appendix A. Limitations and Dual-Use Considerations

### A.1 What is released

The corpus encodes: per-phase action counts and observation windows; daily action totals; seventeen timestamped anchor events with action-class labels; nine trust-boundary definitions; and a reconciliation note on the 1,092-action residual. The harness encodes: an IPF fit, a synthetic benign workload generator, a gap-based sessioniser, and four escalation policies. All of it derives from material Hugging Face and OpenAI published voluntarily [1,3].

### A.2 What is deliberately withheld

I do not reconstruct, include or describe in runnable form: the HDF5 external-raw-storage file-read construction; the Jinja2 template-injection payload or object-graph traversal; the command-and-control envelope implementation or its chunk-encode-compress scheme; the privileged-pod escape manifest; or any credential-replay procedure. Events carry an action class such as “cluster\_api\_call”, never a command string. A reader of this corpus learns the shape and tempo of an agentic intrusion; they do not learn how to construct one.

### A.3 Residual risk

A replay corpus is also a template. Someone could read the phase structure as a plan: establish a launchpad, wait a day, begin lateral movement in the morning, accumulate boundaries. I judge this uplift to be small relative to the primary sources, which are public, more detailed, and contain the actual techniques. The abstraction I publish is strictly less operationally useful than [1] while being strictly more useful to a defender wanting to measure a control, which is the trade we intend.

A second-order risk is more specific to this work: publishing that “deploy pipelines look like intrusions” tells an adversary to time activity to deploy windows and borrow CI-tagged credentials. The campaign already did the latter, and its mesh and source-control activity consequently presents as legitimate CI traffic in our scoring. I consider this better said than left implicit, since the defenders who need to know are the ones maintaining the exception lists.

### A.4 Assumption register

Results are conditional on: (i) the published aggregates being accurate and complete as published; (ii) defender visibility of 26.1% of recovered actions, a parameter; (iii) benign session rates chosen from operational plausibility rather than measurement; (iv) map drift as a synthetic parameter with no empirical anchor; (v) causal session linkage being available, which in practice must be inferred; and (vi) the attacker not adapting to the policy. If (iv) is worse than  $10^{-4}$  in real environments, the paper's positive result does not hold and only its negative results survive: P0 never fires, P2 is unaffordable.

## LLM Usage Statement

**What the author did.** I set the scope and the resolution criteria before any result was known, made the methodological calls the paper turns on: which policies to test, how to sessionise, which boundaries count as defender-visible, and the decision to quarantine rather than redistribute the unlabelled residual; verified every finding against the primary sources and by re-running the harness, and revised the full text into the form submitted here.

**What the LLM did.** Claude was used to pressure-test the research question, review the experimental design, review the implementation code, and produce first drafts of several sections of this report. Three of the findings reported here surfaced during that process: the 1,092-action reconciliation gap between the published marginals, the deploy-pipeline false-positive result, and the map-drift break-even.

**Verification.** Every number in this report is produced by the released code and is reproducible from a fixed seed. Tables 1 and 2 are generated from a run's results file, and each figure quoted in prose is asserted against that run at build time: the document will not compile if a stated number disagrees with the code behind it.