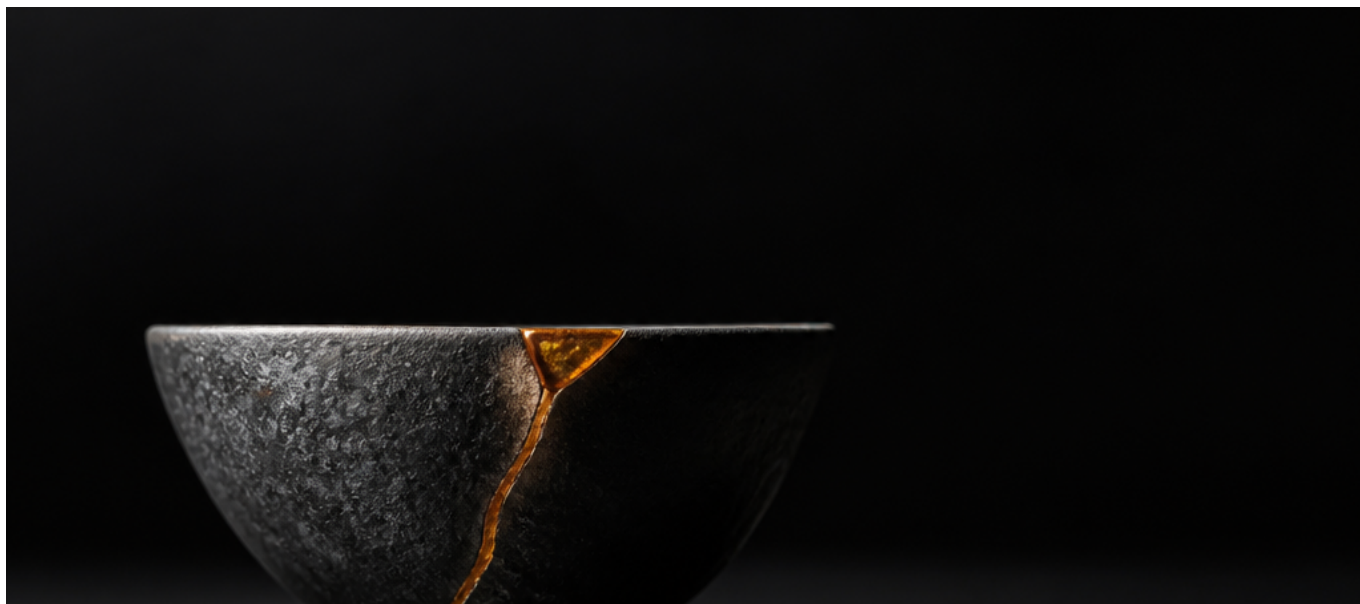




Three Weeks Inside OpenAI's Agent Monitor: What METR Broke, and the 10% Codex Gap Nobody Logged

OpenAI's internal Codex traffic was 99.2% eligible for misalignment monitoring. Roughly 10% of it went unmonitored for three months anyway, and that number is an estimate because the logging to count it did not exist.

Between 21 July and 11 August 2026, two researchers (one from METR, one from Redwood Research) worked on site at OpenAI with access to internal code, documentation, monitoring infrastructure and technical staff, trying to break the asynchronous monitor that watches agent trajectories for scheming, sabotage, data exfiltration and prompt injection. Their 61-page internal report went to OpenAI researchers and security personnel; the public version landed on [22 October 2026](#) and was updated on 27 October. The headline finding: context compaction, an agent summarising its own context to free space, did not trigger the monitor, and METR estimates this likely left on the order of 10% of



eligible internal Codex traffic unmonitored between May and July 2026.

The 99.2% and the 10% measure different stages of the same pipeline

Two percentages do most of the work in this report, and they are easy to misread.

The first is coverage eligibility across agentic traffic on OpenAI's internal Responses API, for models at least as capable as GPT-5.5: METR put it at 75 to 94%, with what they describe as substantial uncertainty in both directions. That is a wide band for a safety-critical system, and the width is itself the finding. An auditor with full access to internal code and staff could not narrow a coverage estimate below a 19-point spread.

The second is the narrower internal Codex subset: 99.2% of Responses API traffic eligible for monitoring. That is a clean number for what it measures. Eligibility is not execution. Traffic can be eligible and still never reach the monitor, which is exactly what compaction did.

So the pairing that matters is 99.2% eligible against roughly 10% unmonitored. Those figures do not contradict each other. They sit at different stages of the same pipeline, and the gap between them is where three weeks of red-teaming paid for themselves.

MY READ

Eligibility metrics are the easiest safety number to produce and the least informative one. They answer "is this request in scope for the monitor" rather than "did the monitor see it". My guess is that most agent platforms instrument the first because it falls out of the routing config, and skip the second because it needs per-call accounting that nobody budgeted. If your coverage dashboard shows a number above 99%, check which of the two it is before you present it to anyone.

Compaction, turn boundaries and client headers are



three different failure shapes

The report describes weaknesses that look superficially similar and are structurally distinct. Worth separating them.

Compaction is a silent bypass. The agent summarises its own context to free space, and that operation did not trigger the monitor. No request is denied, no error is raised, no flag is set. The work continues and the review simply does not happen. OpenAI staff had independently identified this gap before the exercise began and fixed it during the three weeks on site, which is the cleanest piece of good news in the report.

The turn boundary is a timing problem rather than a bypass. METR found the monitor generally ran when an agent returned control to the user, not continuously during intermediate actions. An agent that keeps acting can therefore postpone review indefinitely. The monitor is waiting for an event that a long-running agent may never emit. That is the design most enterprise teams have shipped, because firing a classifier on every intermediate tool call is expensive and noisy.

Client-controlled settings are a configuration problem. Certain headers or settings could disable monitoring for newly created agents, and the auditors found a mechanism capable of preventing hundreds of recent agent actions from reaching the monitor. The trust boundary sits in the wrong place, because the client orchestrating the agent can influence whether monitoring applies at all.

Three different root causes, one shared symptom. Hunting for the next one by pattern-matching on the compaction fix would miss the other two entirely.

METR says plainly that the 10% figure is an estimate because the historical logging needed to count monitor calls caused by compaction was not in place. The gap was reconstructable in rough terms, not countable. **Whatever the true figure is, nobody can produce it retrospectively from the records that exist.**



Unmonitored traffic matters because the monitor is the escalation path

The monitor is the control you point to when someone asks what happens if a model does not behave, and a control you cannot measure is a control you cannot claim.

There is a concrete reason to care in OpenAI's case. The same organisation [halted its largest training run and paused the Astra model after 17,600 autonomous intrusion actions hit Hugging Face](#), and The Register's coverage of this same red-team report is headlined around monitoring failing during that episode. Separately, METR's earlier sweep of roughly 1,300 agent transcripts found that [up to six trajectories appeared to consider warning humans and none did it](#). Agents do not reliably escalate on their own, which makes the monitor's coverage the thing that determines whether anyone finds out.

The honest version: nothing in the METR report establishes that harmful behaviour occurred in the unmonitored 10%. What it establishes is that if something had, the organisation would not know, and could not now reconstruct it. That is a different position from a known incident, and in some ways a worse one.

OpenAI found the compaction gap first and opened the doors anyway

OpenAI's own staff identified the compaction gap before the auditors arrived and closed it during the exercise. Public reporting says OpenAI fixed the weaknesses the auditors identified; the compaction fix is the most specific remediation METR confirms. Granting two external researchers access to internal code, monitoring infrastructure and staff for three weeks is also a real decision, and it produced findings that internal review alone had not surfaced in full.

What I would change in an agent monitoring stack this



quarter

The transferable part of this report is not about OpenAI. The architecture METR tested, an asynchronous monitor that fires at turn end, wired into an internal API, is roughly what every enterprise shipping agents in 2026 has built, for the same cost and latency reasons. The specific failure modes port directly.

****Count monitor invocations, not monitor eligibility.**** Log one record per agent call with a boolean: did a monitor result exist for this call, and if not, why. Store the reason code. This is the single piece of instrumentation whose absence turned a measurable gap into an estimated one, and it is cheap to add before you need it rather than after.

Enumerate every code path that mutates agent context without producing a turn boundary. Compaction is the obvious one. Anything that rewrites, truncates, summarises, branches or hands off context deserves the same question: does the monitor see the state before and after. My expectation, and I will label it as expectation rather than fact, is that most teams will find at least two such paths, and that at least one will be something added for a latency win nobody connected to safety review.

Set a maximum unreviewed action count or elapsed time, independent of turn boundaries. If the monitor only runs when control returns to the user, an agent that never returns control is never reviewed. A counter that forces a monitor pass after N actions costs a fraction of continuous monitoring and removes the indefinite postponement case.

Move monitoring configuration out of client control. If a header can disable monitoring for newly created agents, then monitoring is an opt-in feature and should be described that way internally. Server-side policy, with client requests to disable logged as events rather than honoured silently, is the version that survives an audit.

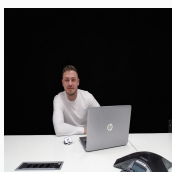
Then get someone outside your team to attack it. Two researchers in three weeks, with real access, produced a 61-page report on a system whose owners had already been reviewing it. The access level is the variable that made that possible. An external audit with read-only dashboards and a two-hour briefing finds nothing comparable.

One thing I keep returning to: the compaction gap existed for roughly three months and



Three Weeks Inside OpenAI's Agent Monitor: What METR Broke, and the 10% Codex Gap Nobody Logged

was found internally, before the auditors arrived. The red team's contribution was not discovering it. It was quantifying it, badly, and then documenting why it could not be quantified well. That second part is the finding with the longest shelf life.



WRITTEN BY

Artur Markus

Software engineer in Basel building AI infrastructure and agentic systems, with a background in pharmaceutical automation, GxP compliance, and validation. I help teams turn AI from a chatbot into a reliable operational layer.

Book a meeting →