MONTH 04 · 1–31 JULY 2026

Two measures improved. *Four went the wrong way.*

Measured against the week-0 baseline of 6 April 2026 by `harness-probe` running inside Contoso Logistics' own CI. No working session was booked in July. This report is unchanged by that.

PILOT SCOPE

`payments-api`, `ledger-svc`, `platform-infra`, `driver-portal` — the last onboarded mid-period on 13 July, which is where most of this month's movement comes from.

PROBE ACCESS

`harness-probe` v0.9.4 ran as a job in your pipelines, under your credentials. No repo write access, no production credentials, no SSO. It reads the harness configuration already present in its own checkout and emits shape, not content — matcher names, scopes, counts, job identities.

ISSUED

Tuesday 4 August 2026.
Next report Thursday 3 September, covering 1–31 August against the same baseline.

Summary verdict

Four months in, the harness holds on the three repositories it was built for and does not hold on the one added last month. Two of six measures improved and four regressed. The two that matter most — full gate coverage at merge, and secret scanning across every repository — both moved backwards, in each case because `driver-portal` was created from an internal template that predates the current gate set.

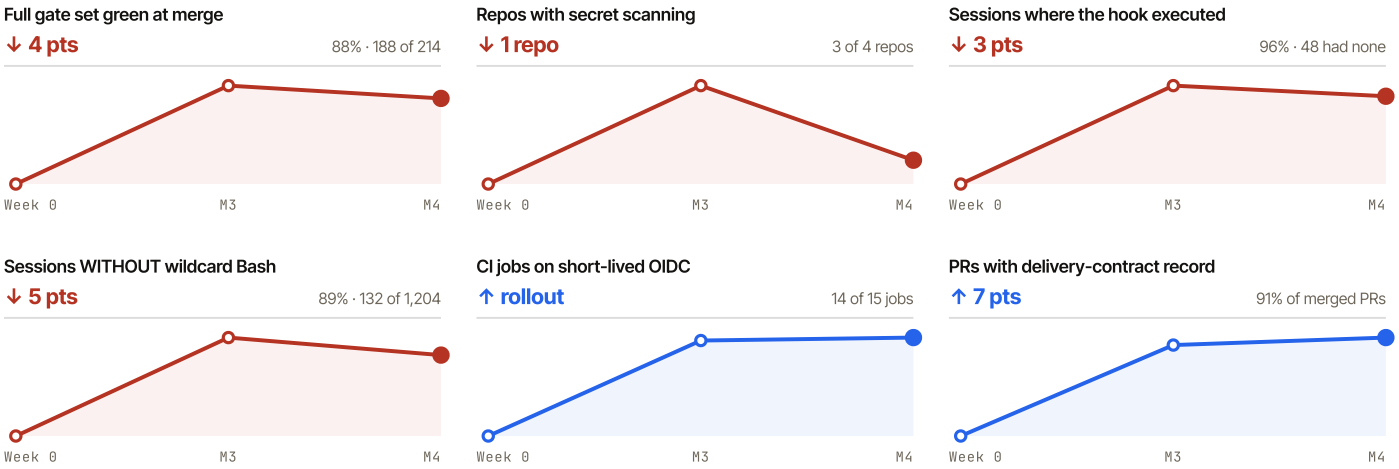
The other two regressions are separate and older: a `PreToolUse` hook removed to unblock a migration and left off for eleven days, and a wildcard `Bash` permission added during an incident that is still in place today.

None of this is evidence of a breach. It is evidence that controls your program assumes are running were not running for part of July — and that the gap closes with configuration changes rather than with a project.

What moves a number here

Every figure in this report is derived from what a job running inside your CI can observe directly: workflow run metadata, job names and outcomes, the resolved `.claude` configuration in its own checkout, `CODEOWNERS`, and the commit range for the run. Nothing is inferred from source code, diffs, or anything held by your source host that the runner is not already given. Where a question cannot be answered from that surface, this report says so rather than estimating — see DR-06 on page 4.

The numbers



Each panel is scaled to its own range, because these measures mix percentages with n-of-m ratios and a shared axis would flatten them into nonsense. Every panel is oriented so that up is better — note the fourth is inverted to "sessions *without* a wildcard `Bash` permission" for that reason. Direction is stated in words as well as colour, so this reads the same in greyscale.

MEASURE	WEEK 0	MONTH 3 JUNE	MONTH 4 JULY	DIRECTION
Merges to `main` with the full required gate set green Semgrep, Trivy, gitleaks, tests	61%	92%	88% 188 of 214	↓ 4 pts
Repositories with secret scanning on every branch	2 of 3	3 of 3	3 of 4 one uncovered	↓ 1 repo
Agent sessions where the `PreToolUse` deny-list hook executed	71%	99%	96% 48 of 1,204 had none	↓ 3 pts
Agent sessions running with a wildcard `Bash` permission	34%	6%	11% 132 of 1,204	↑ 5 pts
CI jobs authenticating with short-lived OIDC rather than static keys	3 of 11	10 of 11	14 of 15	Rollout done
Merged PRs carrying a complete delivery-contract record rule, hook and gate versions	0%	84%	91%	↑ 7 pts

Two caveats on the figures above. The probe container failed to emit between 18 and 20 July — a base-image change on your side removed the CA bundle the uploader needs — so July covers 28 of 31 days. Both the session counts and the merge counts are floors, not totals. And the one remaining non-OIDC CI job is `ledger-svc`'s nightly dependency refresh, which still uses a static registry token issued in 2024.

Findings

DR-01 A `PreToolUse` hook was removed for eleven days on `payments-api`

HIGH

MECHANISM — the `PreToolUse` matcher covering `Bash` was deleted from `.claude/settings.json` on the `release/2026.07` branch on Monday 6 July and restored on Friday 17 July. EVIDENCE — the probe recorded 38 agent sessions on that branch across those eleven days with zero hook-execution events, while session volume itself stayed flat: the hook did not fail, it was absent. Those 38 sessions are most of the 48 behind the drop on page 2. WHY IT LASTED — your own commit message says the hook was blocking a database migration script. That is a real problem, and removing it was a reasonable thirty-minute decision. It lasted eleven days because nothing distinguishes a temporary exception from a permanent one. The branch merged to `main` on 22 July.

DR-02 Secret-scan coverage fell when a new repository was onboarded

HIGH

MECHANISM — `driver-portal` was created on 13 July from the `svc-template-node` internal template, last updated before gitleaks joined the standard gate set. The template's workflow includes Semgrep and Trivy and omits any secret-scan step, so the new repository inherited two of three scanners and nobody had reason to look. EVIDENCE — 22 merges to `main` between 13 and 31 July with no secret-scan job in the run; the figure excludes the 18–20 July probe gap. LIMIT — the probe cannot tell you whether a secret actually landed there, because it never reads file contents. What it can tell you is that if one had, nothing was watching. THE REAL FINDING — the template, not the repository. Your platform team confirmed eleven repositories have been created from it since January; this is simply the first one inside the pilot.

DR-03 A permission scope widened during an incident and never narrowed

MEDIUM, TRENDING HIGH

MECHANISM — `Bash(*)` was added to the `permissions.allow` list in `ledger-svc`'s `.claude/settings.json` on 9 July, in a pull request titled "unblock incident tooling" that merged the same day. It is still there as of 4 August. EVIDENCE — wildcard `Bash` sessions across the pilot rose from 6% in June to 11% in July, and 118 of those 132 sessions are on `ledger-svc`. WHY IT IS EASY TO MISS — a permission widening is a four-word diff in a config file, reviewed at 2am, inside a change everyone wanted merged. It does not look like a security change and no gate treats it as one. The incident it was opened for closed on 10 July. Your own tool-call logs hold the commands those sessions actually invoked, which is the list to rebuild the allow-list from.

DR-04 Two merges bypassed the gates, and nothing re-ran them afterwards

MEDIUM

MECHANISM — two merges to `payments-api` `main`, on 22 and 29 July and both tied to the same release, completed without the required gate jobs appearing in the run at all. EVIDENCE — the probe sees the commit range and the jobs that ran against it; for these two commits, no Semgrep, Trivy, gitleaks or test job exists. It cannot see *which* mechanism let them through — an admin override, a ruleset exemption, or a misconfigured required-check list are all consistent with what it observed, and your source host can tell you which. THE GAP — whichever it was, neither commit was re-scanned afterwards, so two commits sit in that history today that no scanner has ever seen. We are not recommending you remove break-glass; teams that remove it invent something worse. We are recommending it carry an expiry.

DR-05 Two MCP servers were added without a review step

MEDIUM

MECHANISM — `.mcp.json` in `driver-portal` and `ledger-svc` each gained one server in July. The probe reports server name and transport only, never environment values or arguments. One is launched via `npx` against an unpinned package specifier, which resolves to whatever the registry publishes at session start. EVIDENCE — no `CODEOWNERS` entry covers `.mcp.json` in either repository, so both changes merged on a single approval from inside the owning team. ASSESSMENT — nothing has gone wrong. But an MCP server is an unreviewed integration holding tool access, and an unpinned one is a dependency you re-select every morning.

DR-06 We should have seen DR-02 coming, and did not

CORRECTION

Last month's report noted that `driver-portal` was scheduled for onboarding in July and treated it as a scope note. We did not ask which template it would be created from, and we did not check whether that template carried the current gate set — both answerable in April, from data we already had. CONSEQUENCE — a regression we could have prevented with one question became a finding we reported after the fact, and 22 merges went through an unscanned repository in between. CHANGE — from next month, any repository scheduled to enter the pilot gets its template checked against the gate set before onboarding, and that check appears in the report whether or not it finds anything. A drift report that only describes the past is worth less than one that occasionally prevents an entry in it.

DR-07 Not measurable: whether agent-authored changes were actually reviewed

OPEN LIMIT

This is the question you asked in month two and we still cannot answer it. The probe sees that a merge's required approval check passed and that a delivery-contract record was attached. It cannot see whether the approving engineer read the diff, understood it, or approved a 900-line agent-authored change in the time it takes to read the title. Answering that honestly needs review-latency and diff-size data from your source host, which means read access to your repositories — and the premise of this arrangement is that we do not have it and are not asking for it. WHAT WE WOULD SAY INSTEAD — 91% delivery-contract coverage tells you the machinery recorded who was accountable. It tells you nothing about whether accountability was exercised. Do not let anyone read that number as a review-quality metric, ourselves included. If you want it measured, it is a job for something inside your perimeter, and we will help you specify it.

Actions, in priority order

1	Restore the `PreToolUse Bash` matcher on `payments-api` and make its presence a required status check, so removing it fails CI rather than passing quietly. Fixes DR-01 and the class it belongs to.	**Platform eng lead** ~1 day
2	Add the gitleaks step to `svc-template-node` first, then to `driver-portal` and the ten other repositories built from it. Template first, or this recurs next quarter.	**AppSec engineer** 0.5 day + 2–3 days backfill
3	Replace `Bash(*)` on `ledger-svc` with an explicit list of the commands the incident actually needed, rebuilt from your tool-call logs.	**team-borealis lead** 2 hrs + 1 wk watching
4	Give gate bypass an expiry: any merge without the required jobs opens a ticket automatically and re-runs the full gate set against `main` within 24 hours. Backfill the two July commits this week.	**Release manager** ~3 days
5	Pin the MCP server package to a version and add `.mcp.json` to `CODEOWNERS` in all four pilot repositories, so a new server needs review from outside the owning team.	**Platform eng lead** ~1 day
6	Decide whether review depth is worth instrumenting at all, given it must be built inside your perimeter. A decision, not a project — see DR-07.	**VP Engineering** One 45-min conversation

What this sample is

Contoso Logistics does not exist. The company, the repositories, the teams, the dates, the incidents and every number in this report are invented. What is not invented is the format, the measures, the mechanisms, and the willingness to report a bad month: four of six measures here move the wrong way, DR-06 is us correcting our own analysis from the month before, and DR-07 is a question we cannot answer without access we have promised not to ask for.

A real customer's report is confidential and will never appear on our site or in a deck — which is exactly why this one was written from scratch. If a sample artifact reports that everything is fine, it is not evidence of anything.

Items 1 through 3 close the four regressions on page 2. If nothing else on this list gets done, do those. Next month's report will show whether they held, and will say so plainly if they did not.