

Agents are now arguing in your pipelines.

What 4,007 developer replies to Corgea findings on pull requests, posted between October 2025 and September 2026 across our customer base, reveal about who is triaging application security now.

0% → 49%

of replies written by an AI agent, October 2025 to August 2026

19.5%

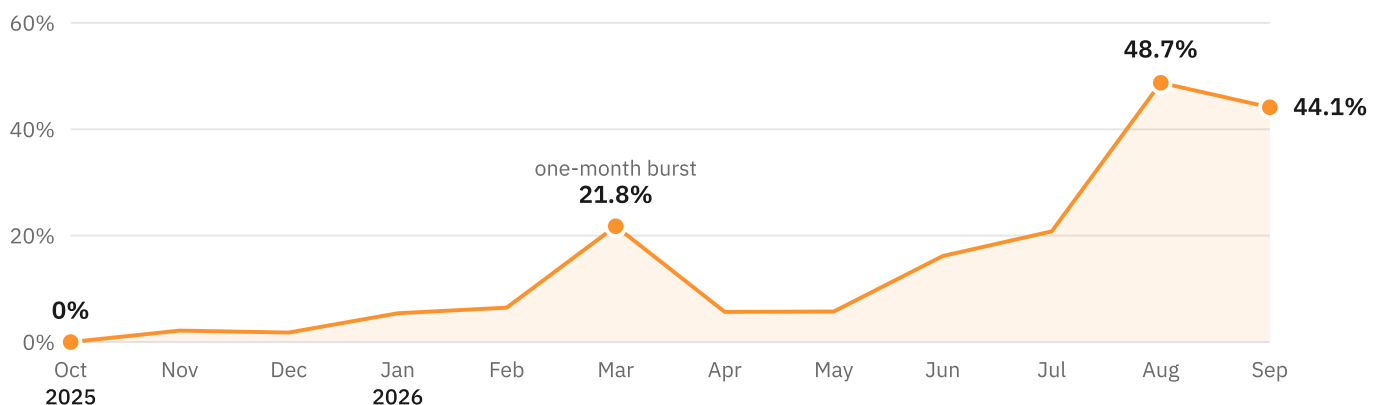
of all replies across the year were agent-written (95% CI 16.7–22.3)

5.6×

the odds that an agent reply accepts the risk instead of fixing it, compared with a developer reply

Agent-written replies as a share of all replies, by month

Default classifier threshold. Replies per month range from 24 (Oct 2025) to 1,015 (Mar 2026). Odds of an agent-written reply grew 1.33× per month.



Why this matters

Findings have always been written for a person: a short title, a severity, a suggested patch. Nearly half of the replies we now receive come from a coding agent that reads the finding, inspects the repository and answers on the developer's behalf. Agents do not ask what a finding means or how to fix it. They rule on it, cite the code and move on. A security program that assumes a human reads every alert now has a second reader. This brief describes how it behaves.

AI replies argue. Developers report status.

Developers answer a finding with one or two words: done, fixed, false positive. Agents answer with a verdict. Disputes, risk acceptances and deferrals together make up 60% of agent replies, and an agent is 5.6× more likely than a developer to accept the risk rather than change the code. Questions and instructions to the scanner, the two most human replies, almost never come from an agent.

What the reply does

Share of replies within each author class

■ AI agent replies (n=763) ■ Developer replies (n=1,433)



Verdicts, meaning a dispute, a risk acceptance or a deferral, make up 60% of agent replies.

The argumentation gap

Among replies that dispute the finding

49.8% vs **0.6%**

cite hard evidence: a file and line, a commit hash, a test run, a code block, a URL or a specification. 35.6% of agent disputes point to a file and line; none of the 341 developer disputes do.

62 vs **7**

median words per dispute. The developer's dispute is an assertion; the agent's is a case.

AI agent reply disputing a finding

False positive — this repo is on React 19.2.5 (see `pnpm-workspace.yaml`), where `ref` as a plain prop on function components is the officially supported pattern...

Cites the lockfile, the framework version and the pattern's use elsewhere in the codebase.

Developer reply disputing a finding

false positive

One of 47 identical replies with this exact text. Half of all developer replies (50.2%) are exact duplicates of another reply; 16.4% of agent replies are.

AI agent reply accepting the risk

Acknowledged. The `TRY_CAST(ErrorID AS INT)` approach is intentional — it handles source data where `ErrorID` can be non-numeric.

Names the construct, states the design intent, closes the thread. No human is asked.

Developer reply accepting the risk

Accepted risk

Risk acceptance is 3.7% of developer replies and 17.7% of agent replies.

Where agents show up, and how they behave.

Teams point agents at high-volume, low-drama findings first. When an agent does answer a classic security finding it is far more likely to argue than to fix, and it does so in bursts: dozens of distinct, evidence-backed verdicts inside the same minute.

Used three times as often on code quality, arguing twice as hard on security

Share of replies written by an agent



Share of agent replies that dispute the finding



Code quality: always-incorrect control flow, dead code, empty catch. Classic security: XSS, SQL injection, missing authorization, IDOR, open redirect, CSRF. Agents dispute IDOR, SQL injection and missing-authorization findings 60–71% of the time.

Fingerprints agents leave behind

- 40** replies carry an explicit "Generated with Claude Code" attribution footer. Every signed reply names Claude Code.
- 13** carry terminal copy-paste artifacts (lines padded with trailing spaces)
- 5** cite the repository's CLAUDE.md or AGENTS.md conventions file
- 3** leak a `/tmp/claude-.../scratchpad` path instead of the reply itself
- 42%** of agent replies open with a verdict formula ("Refuted —", "Not applying this one —", "Won't fix —"), versus 10.3% of developer replies

They arrive in bursts

0 min

median gap between an agent reply and the previous reply in the corpus, versus 3 minutes for developer replies

51.6%

of agent replies land in the same minute as the reply before them

49 in 11

verdicts in eleven minutes: the largest single agent session, 38 distinct texts across four vulnerability classes

Agents push back on the fix, not just the finding

38 of 44

explicit refusals of the scanner's suggested patch are agent-written

15 of 19

replies telling the scanner it cited code that does not exist are agent-written

- AI agent reply declining a patch

Not applying this one — the loop terminates, and the suggested patch would break the script.

What this means for your AppSec program

Findings must be verifiable by an agent.

The reader checks the file, line and data flow you cite. A finding that names code that is not there is refuted within the minute.

Watch automated risk acceptance.

Agents accept risk at 5.6× the developer rate and post verdicts in batches. Your policy on who may accept a risk must cover an agent acting for a developer.

Treat agent disputes as evidence to check, not noise.

Half cite hard evidence. Read the case, confirm the control it names, and feed confirmed ones back into scanner rules.