Developer

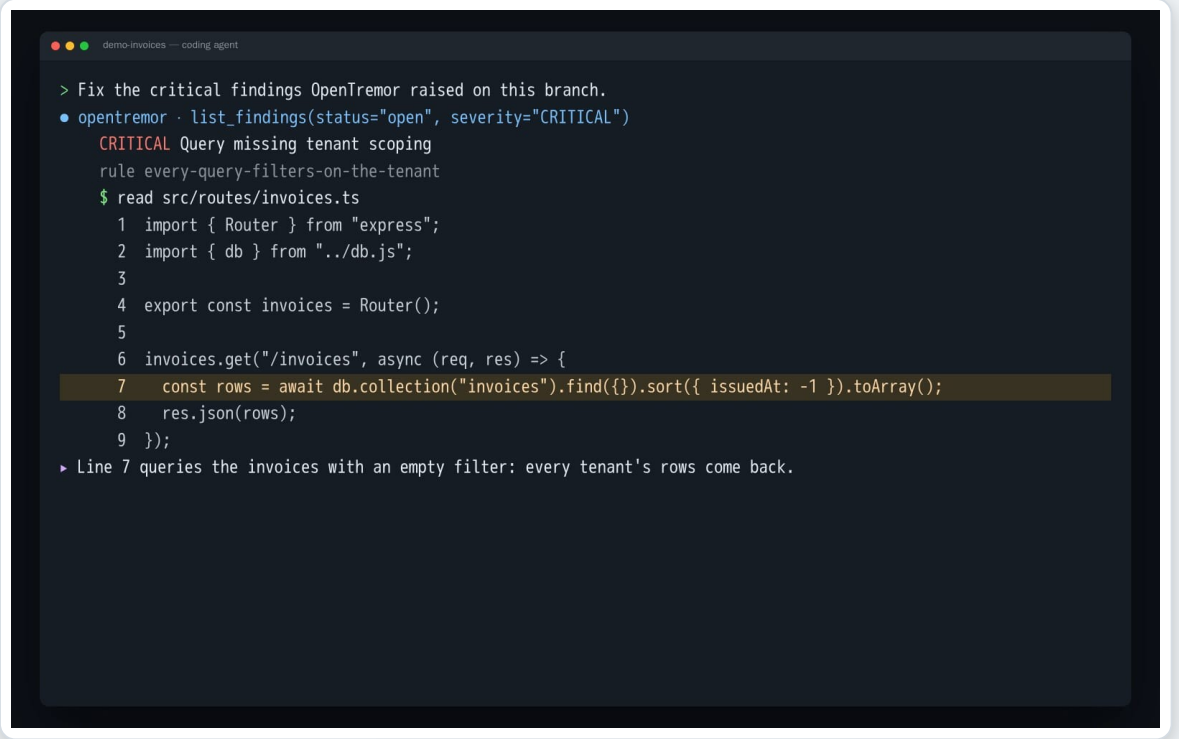


Findings you can fix from your editor, with your coding agent.

From finding to fix, with your coding agent

“As a developer, I want my coding agent to read OpenTremor's findings and fix them, so I do not have to copy findings into a prompt.”

Findings are in a dashboard. The agent that writes the code never sees them, unless someone copies them into a prompt.

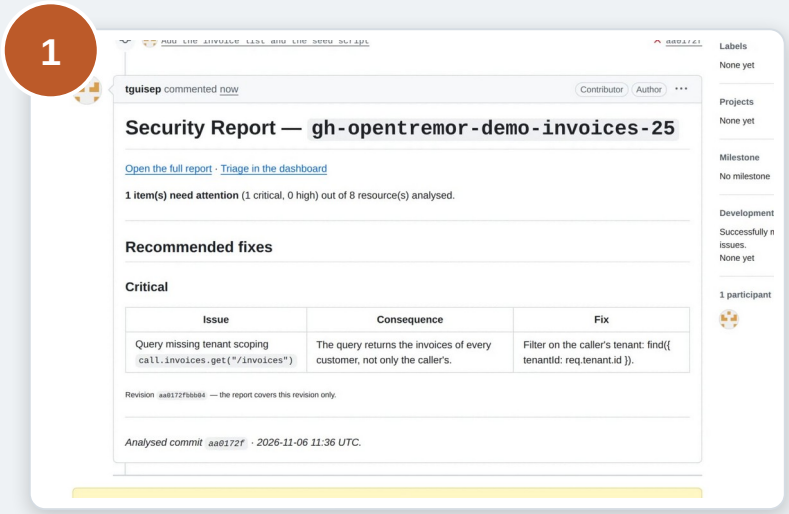


What to notice

- OpenTremor is part of the agent's loop: OpenTremor finds, the agent fixes, and the next review checks the fix.
- The agent uses your access token. By default, it can only read.
- The calls to OpenTremor are real. The agent's words and its edit are scripted for the recording.

From finding to fix, with your coding agent

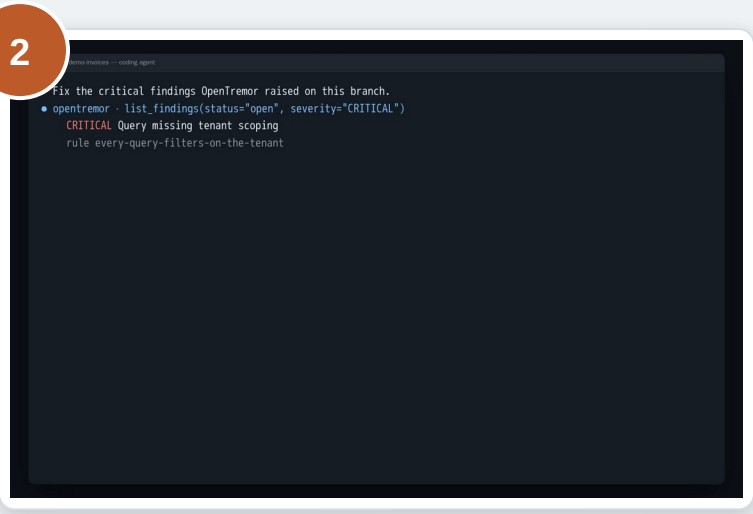
1



OpenTremor flags a critical finding on the pull request.

OpenTremor's comment on the pull request lists a critical finding: the invoice query reads every customer's data.

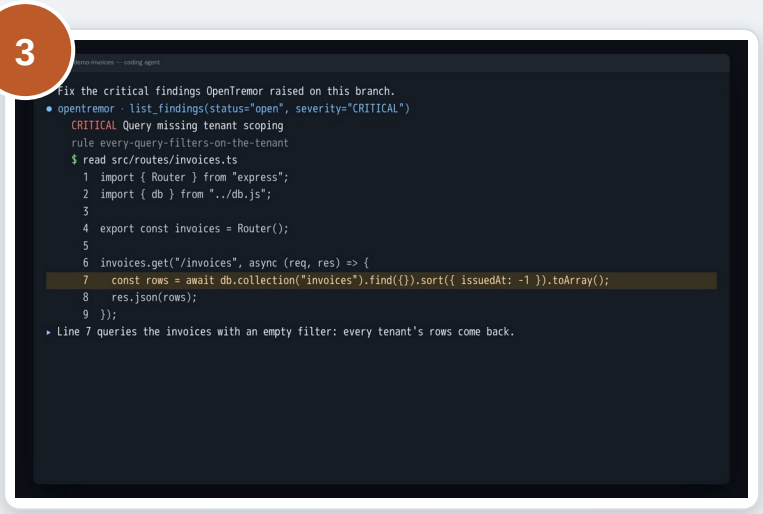
2



Your coding agent fetches it through MCP.

The developer asks the agent to fix it. The agent asks OpenTremor for the open findings (list_findings, over MCP) and gets the finding and its rule.

3

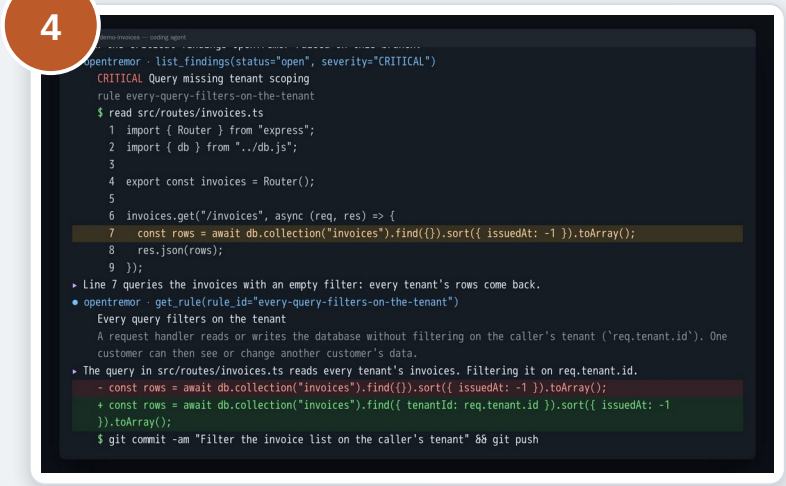


It reads the code the finding points to.

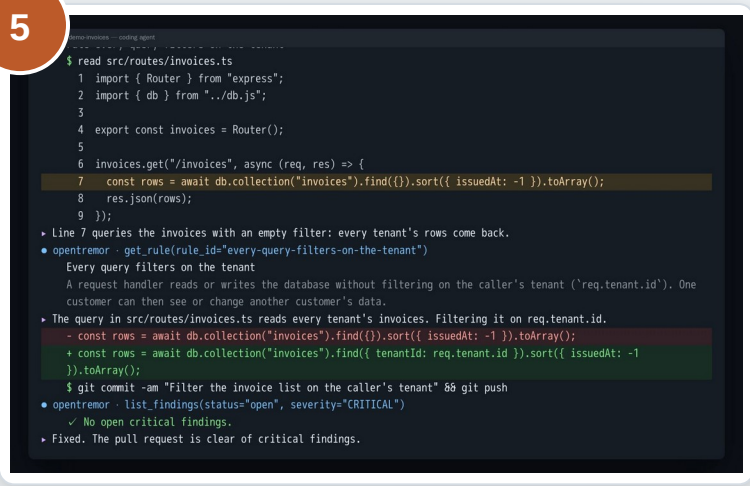
The agent reads the file the finding points to. Line 7 reads the invoices without any filter.

From finding to fix, with your coding agent

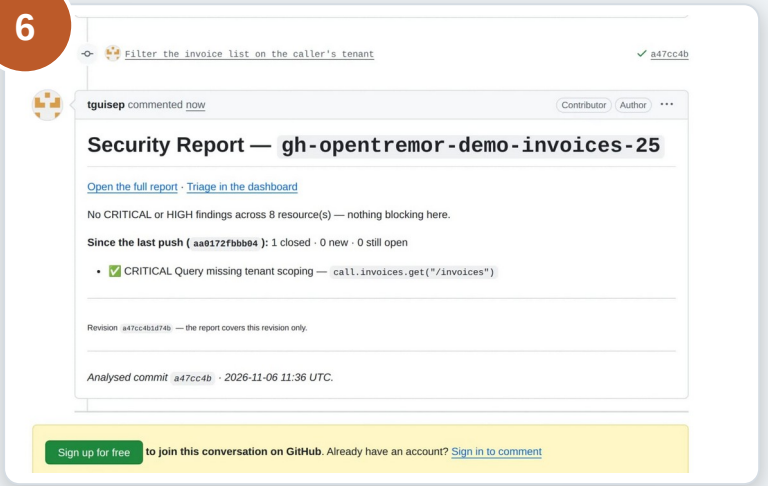
4



5



6



It reads your team's rule, fixes the code and pushes.

It reads the team's rule (get_rule). Then it adds the missing filter and pushes the change.

OpenTremor reviews the new commit. Nothing is left open.

OpenTremor reviews the new commit. The agent asks again, and no finding is open.

The finding is closed, and the pull request records it.

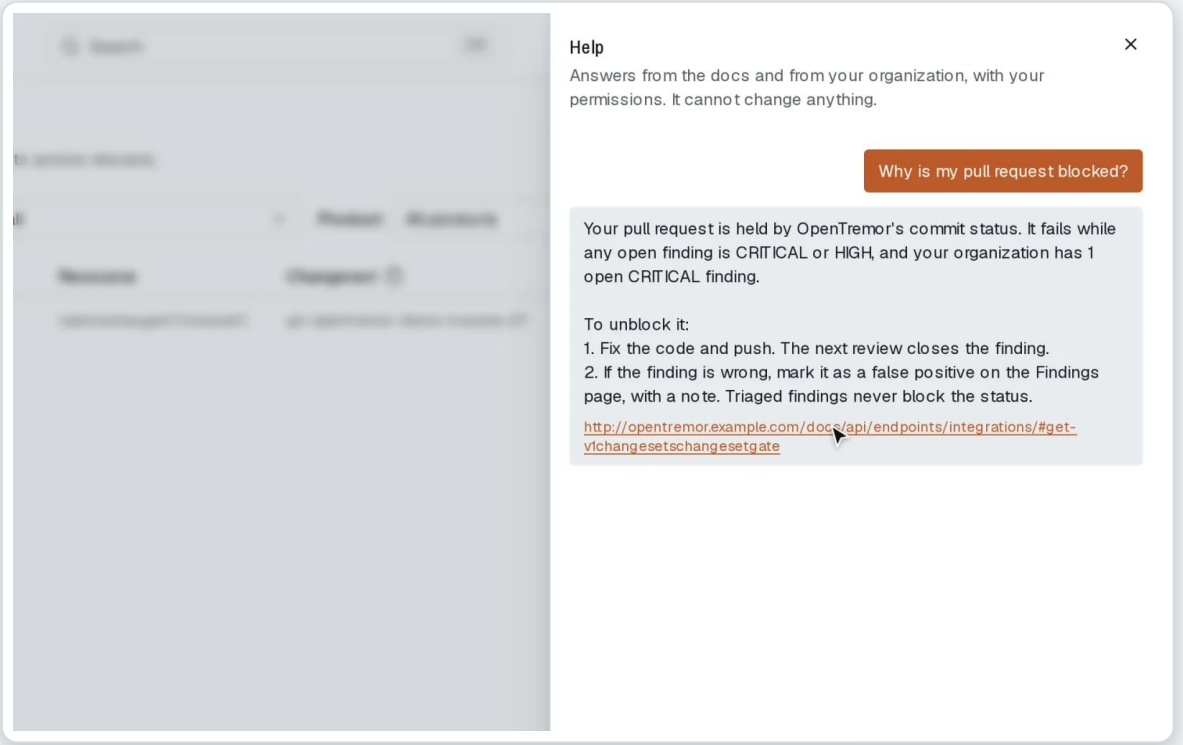
OpenTremor's comment shows that the finding closed after the last push. The pull request keeps the record.

02 · ANY TEAM MEMBER

Ask why a pull request is blocked

“As a developer, I want to ask why my pull request is blocked and get an answer from our own data, so I do not have to wait for the platform team.”

A red check says that something failed. To find out why, you read the docs or ask an expert.

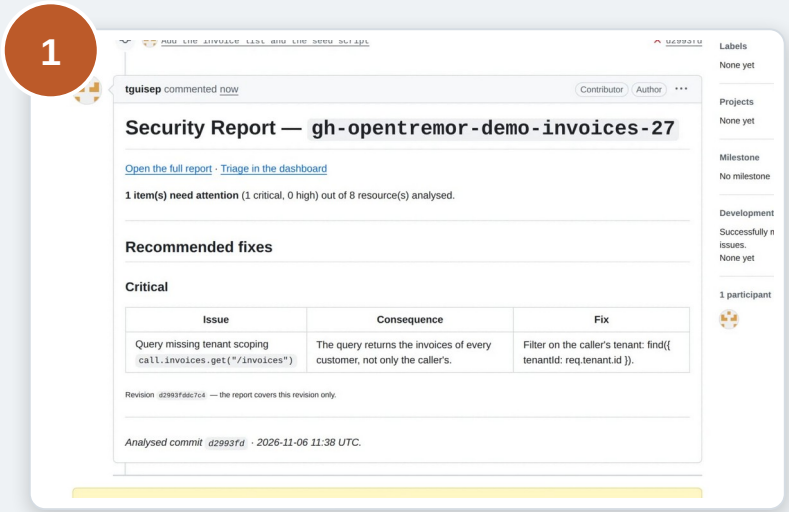


What to notice

- An organization admin turns the assistant on. It uses the organization's own AI model connection.
- Each answer links to the docs it uses.

Ask why a pull request is blocked

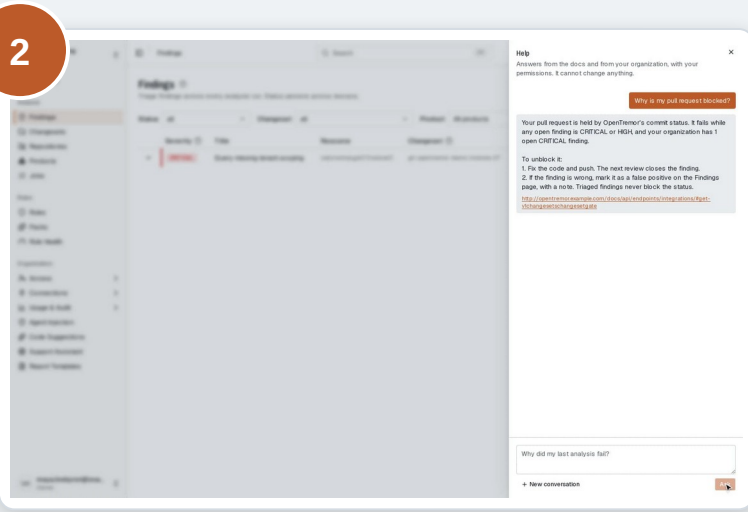
1



A pull request is blocked.

OpenTremor's comment shows one critical finding, and the check fails.

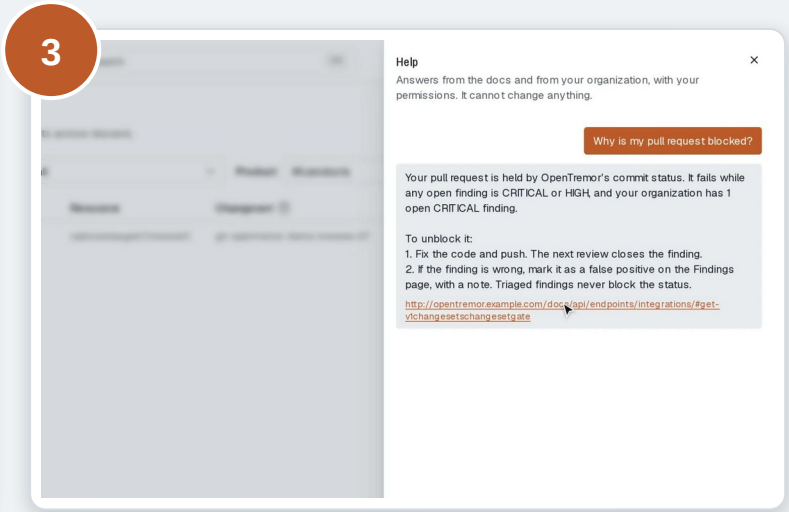
2



The developer asks the assistant, from any page.

The Help button on every dashboard page opens the assistant. It has the developer's permissions and cannot change anything.

3



It checks your findings and the docs, and says how to unblock it.

It reads the organization's findings and the docs. It explains what blocks the pull request, gives two possible fixes, and links to the docs page.

03 · REVIEWER

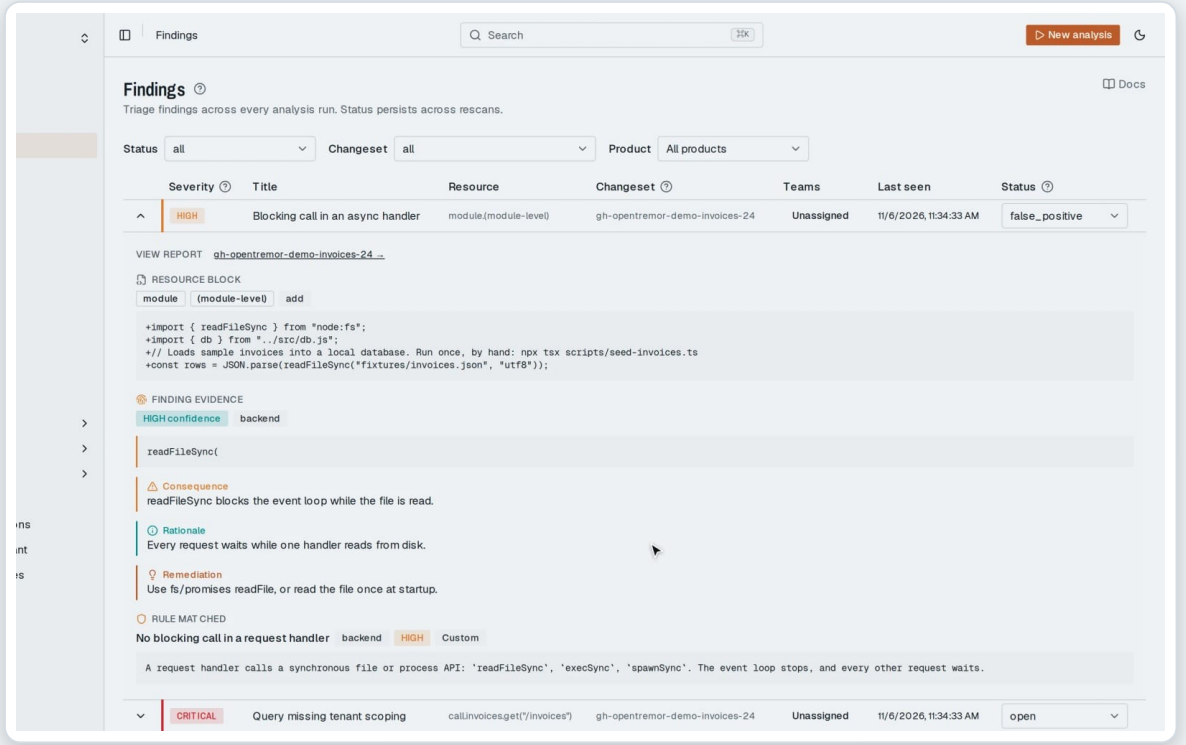
Dismiss a false positive once

“As a reviewer, I want to dismiss a false positive with a reason, so it does not come back after every push.”

Most review tools forget. The same false positive comes back on the next commit, and people start to ignore the tool.

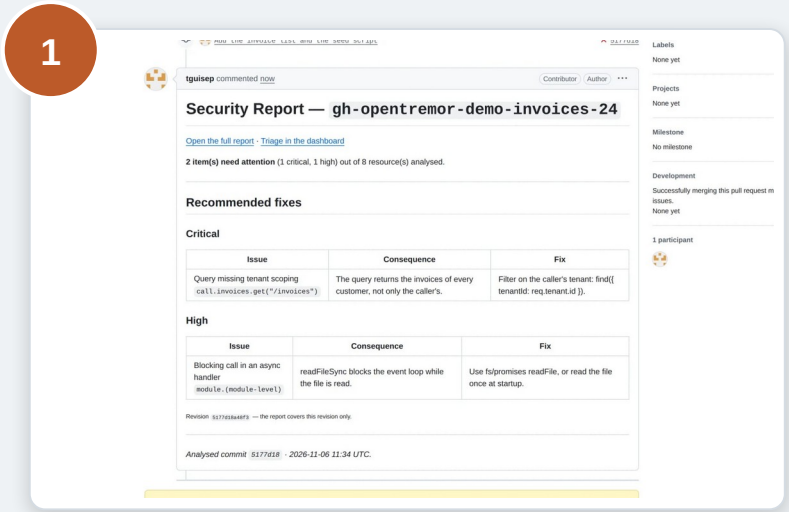
What to notice

- The note stays with the finding, for the next reviewer and for audits.
- Each decision counts against the rule. When reviewers often reject a rule, OpenTremor suggests a change to it.



Dismiss a false positive once

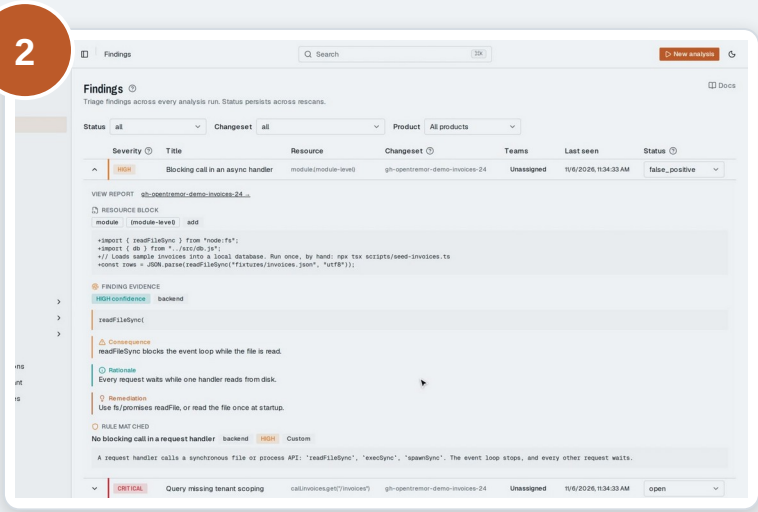
1



Every pull request is reviewed against your rules.

OpenTremor's comment on the pull request lists two findings: the customer data bug and a slow, blocking call in a script.

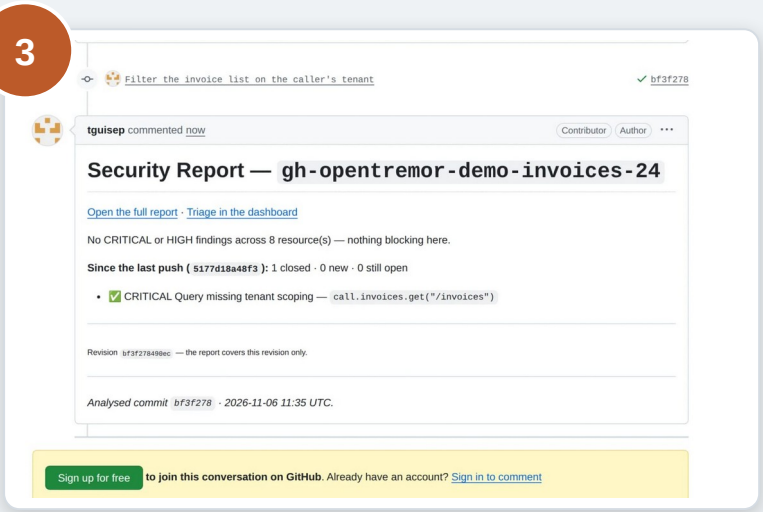
2



A reviewer dismisses a false positive once, with a reason.

The script runs only once, by hand, so the blocking call is not a problem. The reviewer marks it as a false positive and writes why.

3



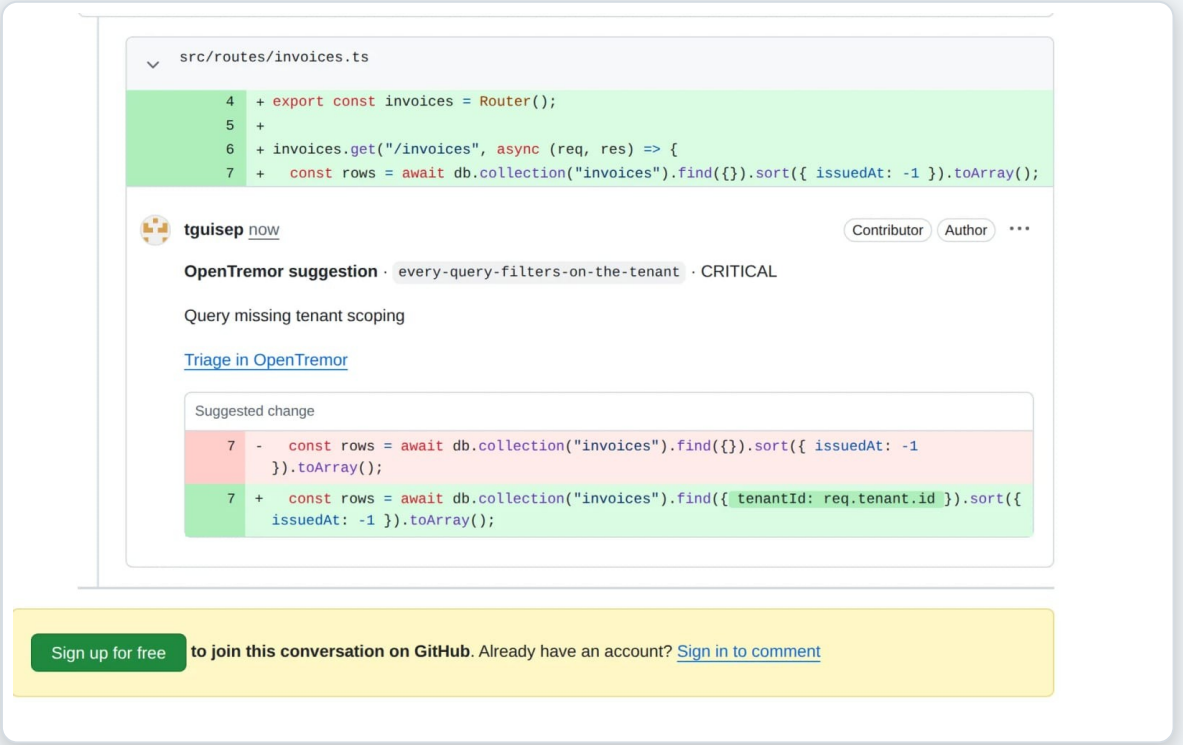
The next push: the fix closes the finding, and the dismissal stays.

The developer pushes the fix for the customer data bug, and that finding closes. The dismissed finding stays dismissed, with its note.

Every pull request checked against your team's rules

“As a tech lead, I want every pull request checked against our written standards, so reviewers stop repeating the same comments.”

Your standards are in a wiki and in people's heads. A pull request follows them only if the reviewer knows them and has time.



What to notice

- The rule is your team's own, written in plain words. You do not need a special rule language.
- The finding explains the risk, not only the line of code.
- The fix is a normal GitHub suggestion. OpenTremor never commits code for you.

Every pull request checked against your team's rules

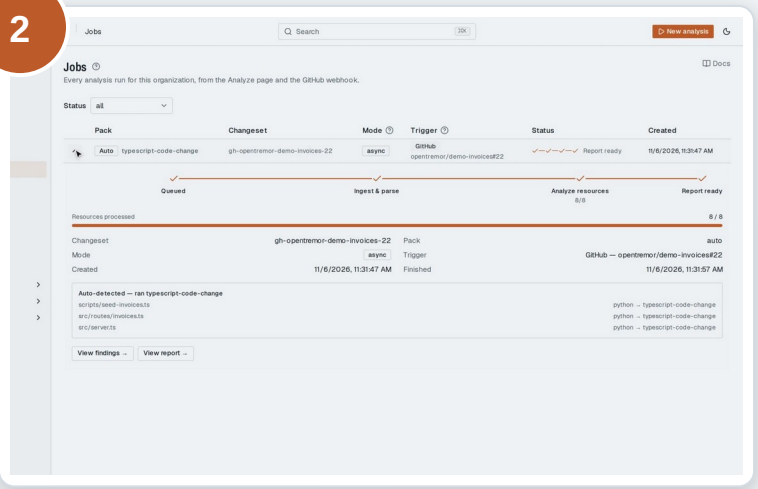
1



A developer opens a pull request.

The pull request adds a list of invoices. Its database query returns the invoices of all customers, not only the caller's.

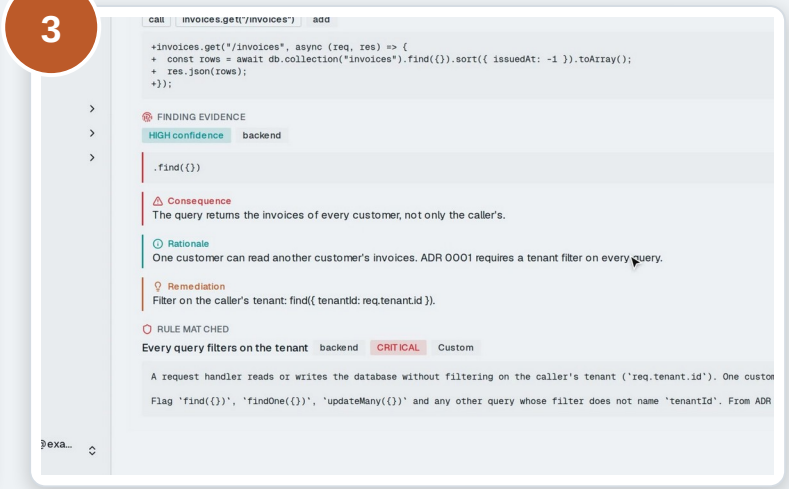
2



OpenTremor reviews it against your team's rules.

OpenTremor receives the pull request and checks the changed code against the organization's rules.

3

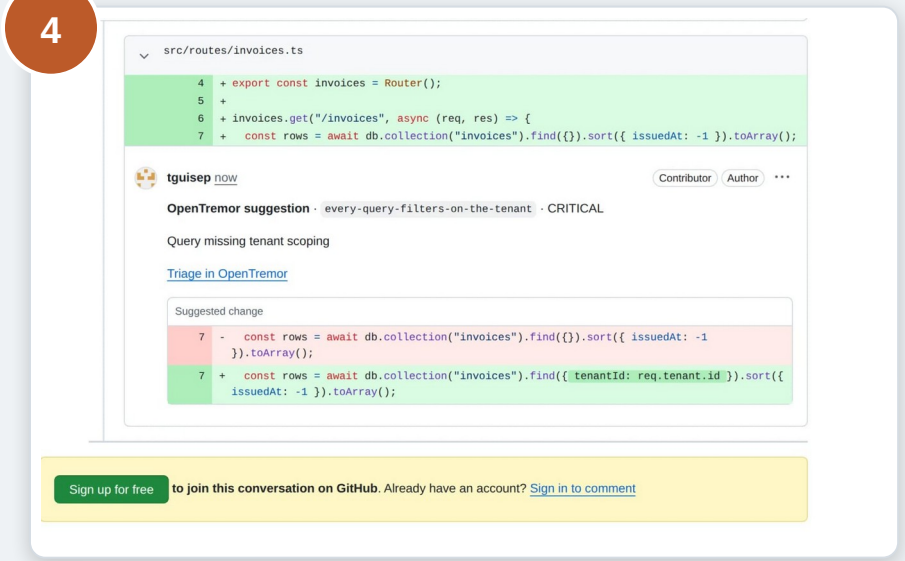


Each finding names the rule it breaks and why it matters.

The finding names the broken rule, "Every query filters on the tenant". It also explains the risk if the code goes to production.

Every pull request checked against your team's rules

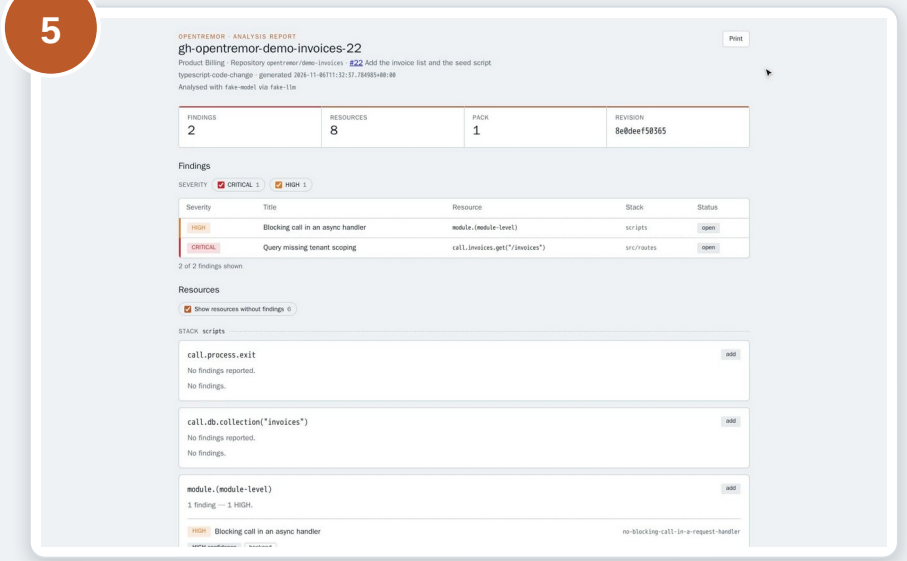
4



The fix comes back to the pull request, ready to commit.

For a critical finding, OpenTremor suggests the fix on GitHub. The developer accepts it from the pull request.

5



The full report, shared with one link.

The report covers the whole change. Anyone with the link can read it until the link expires.

NEXT

Read more

Agents

Connect Claude Code, Cursor or another MCP client.

opentremor.dev/agents

Dismiss a false positive

Step by step, with screenshots.

docs.opentremor.dev/guide/triage/

Developer docs

Triage, suggestions, agents and the API, in order.

docs.opentremor.dev/personas/developer/

See it in motion



opentremor.dev/tour/for/developer · docs.opentremor.dev