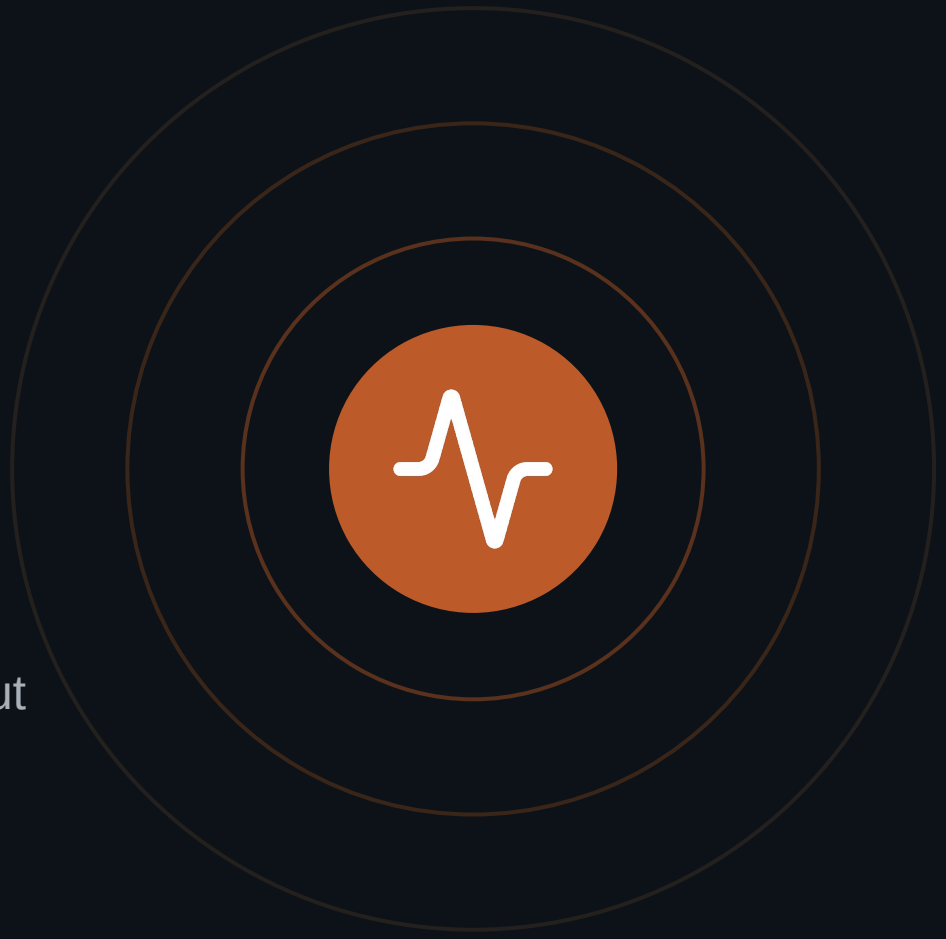
Tech lead

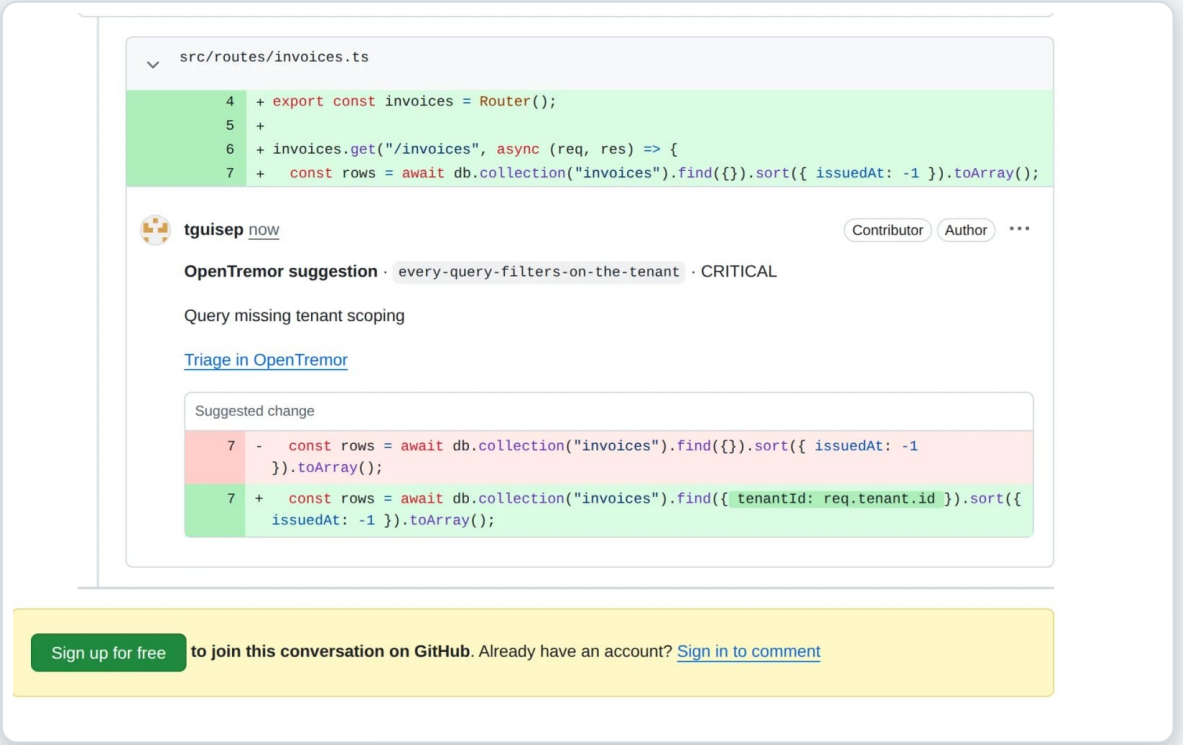


Your team's rules checked on every pull request, so reviews talk about the design.

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.

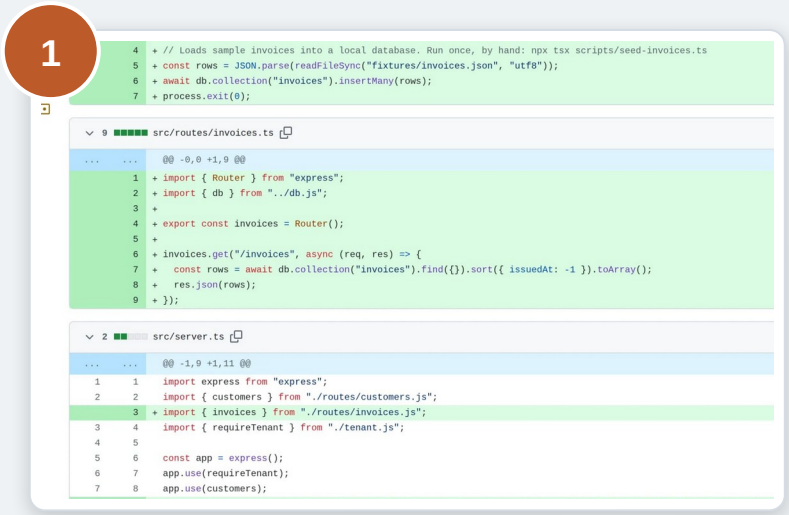


What to notice

- 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

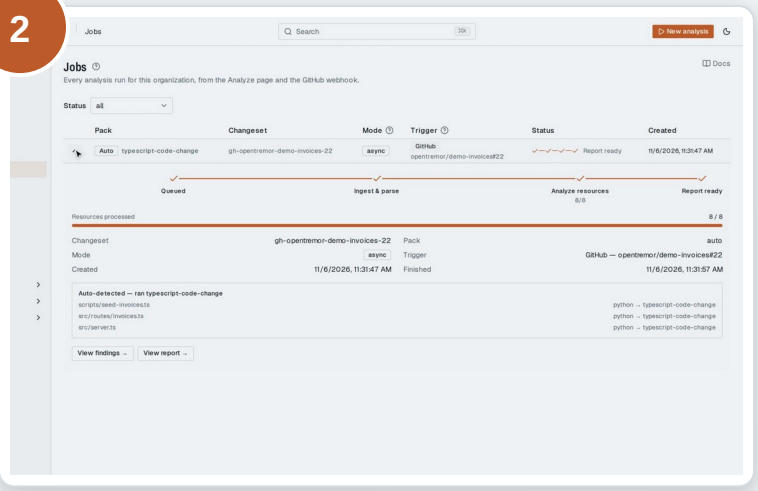
1



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.

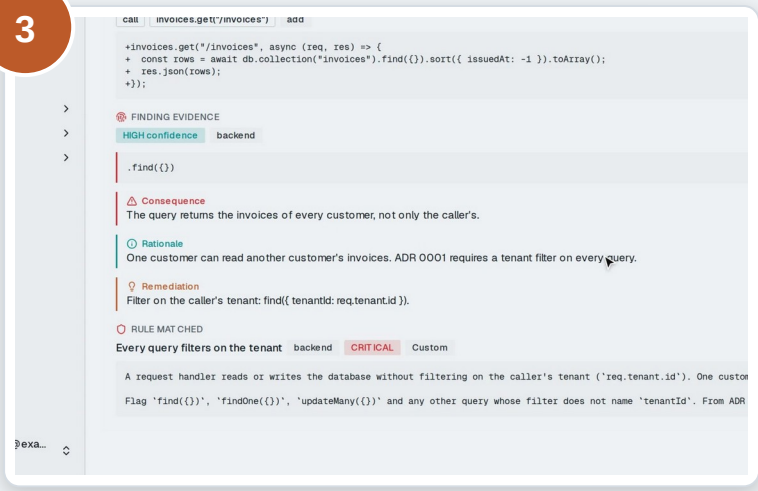
2



OpenTremor reviews it against your team's rules.

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

3

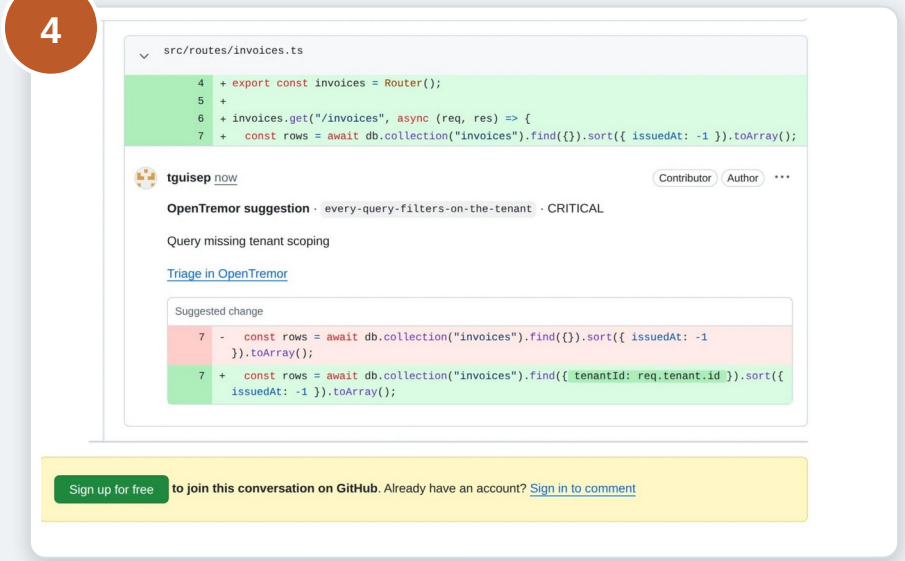


Each finding names the rule it breaks and why it matters.

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

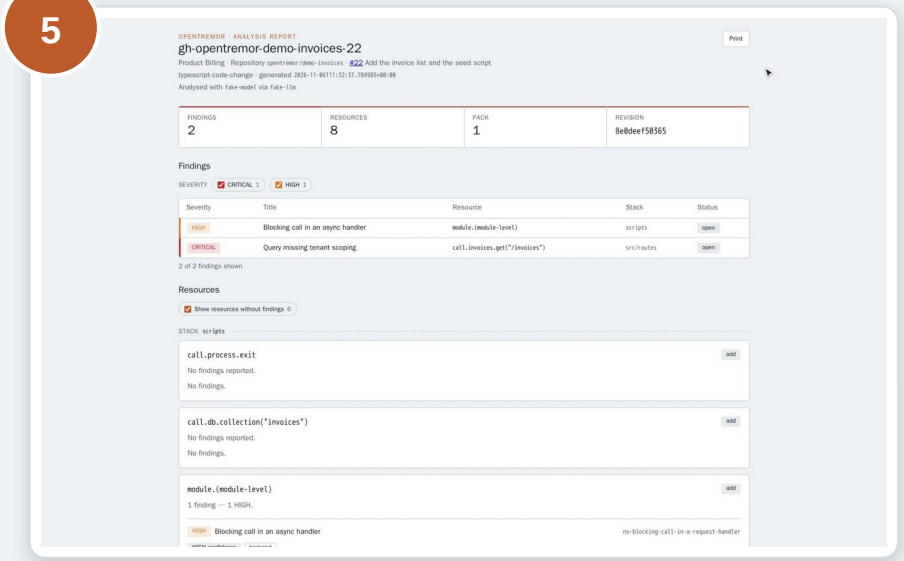
4



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.

02 · REVIEWER

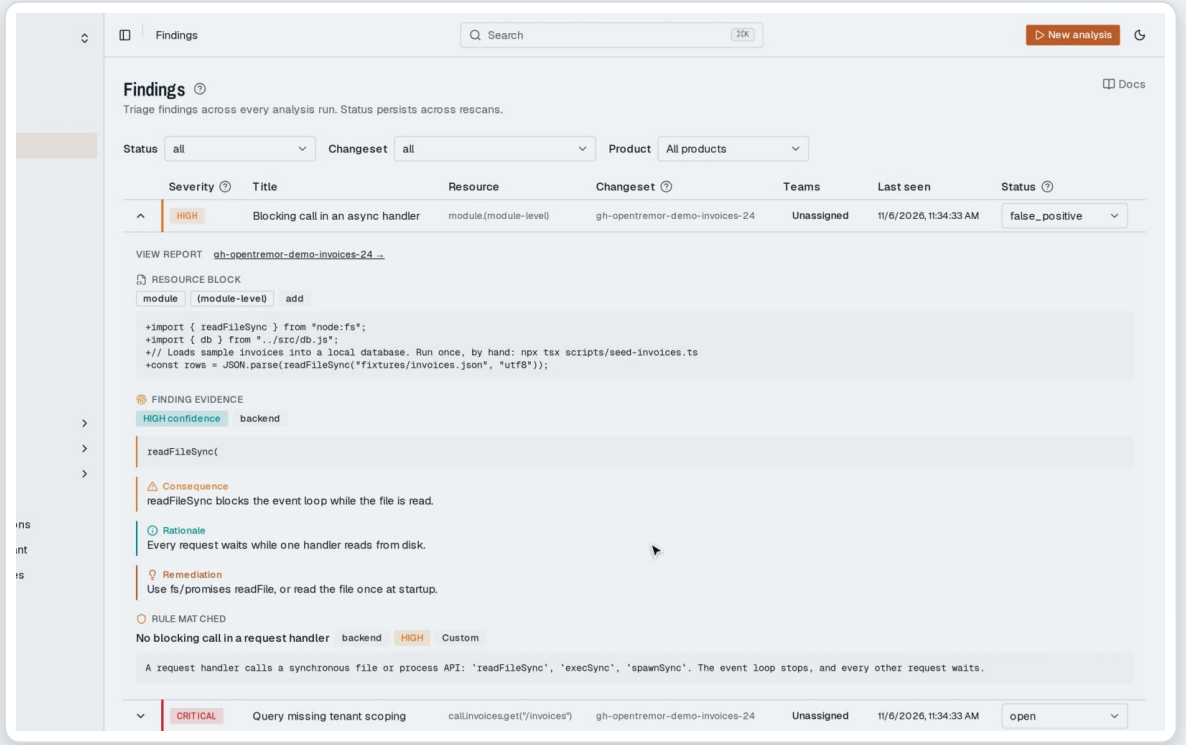
Dismiss a false positive once

“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.

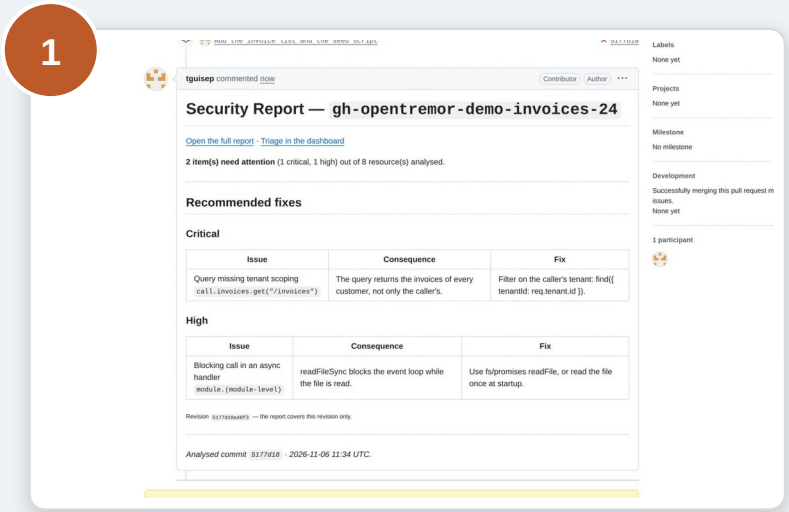
What to notice

- 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

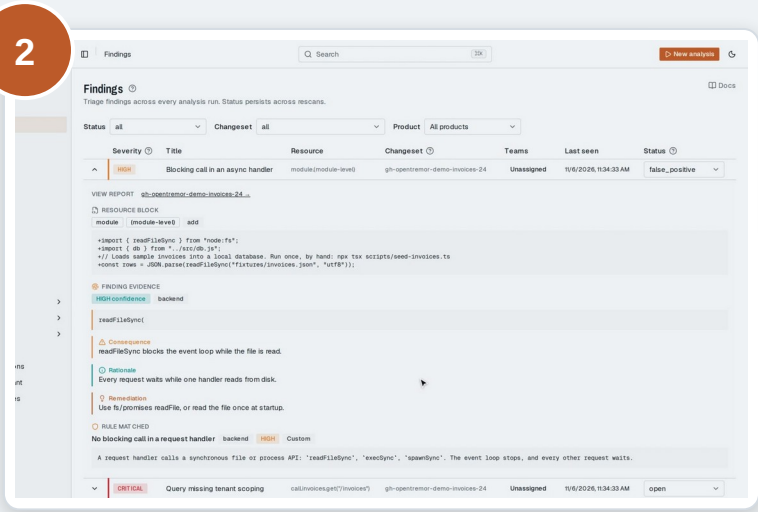
1



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.

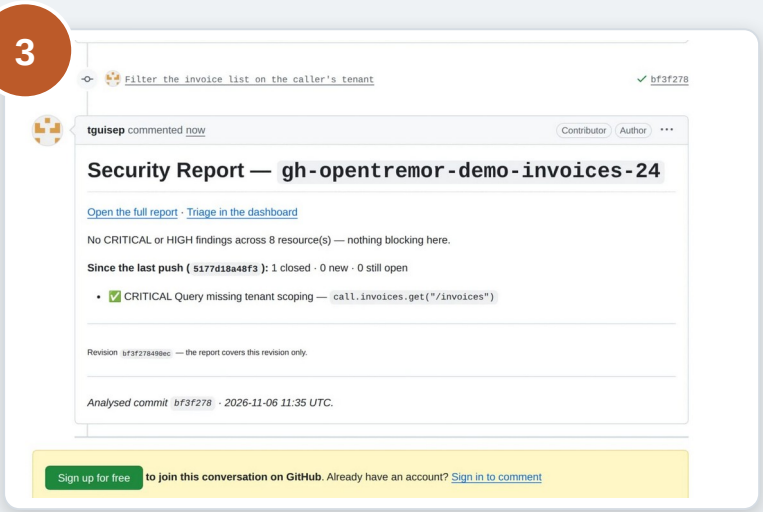
2



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.

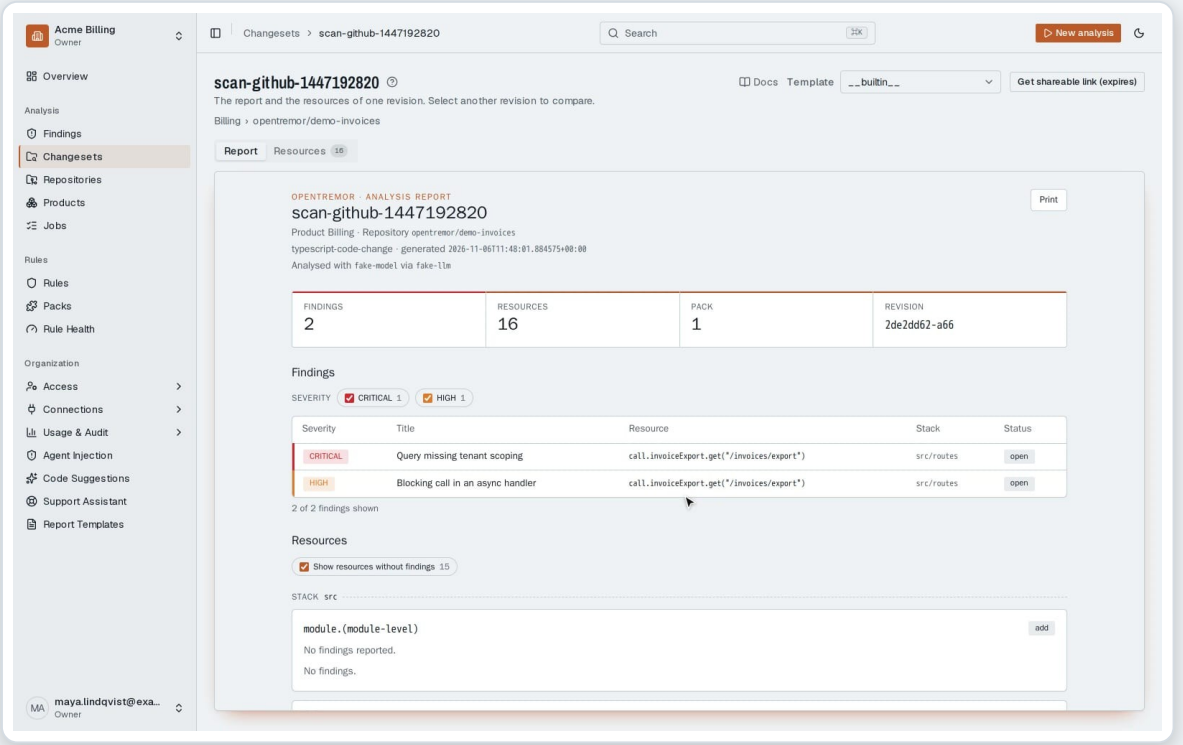
Find problems already in your code

“As a platform engineer, I want to check an existing repository against our standards, so I can find its problems before they cause an incident.”

Pull request review sees only the changed code. Nobody checks the older code, so nobody knows how much of it breaks the rules.

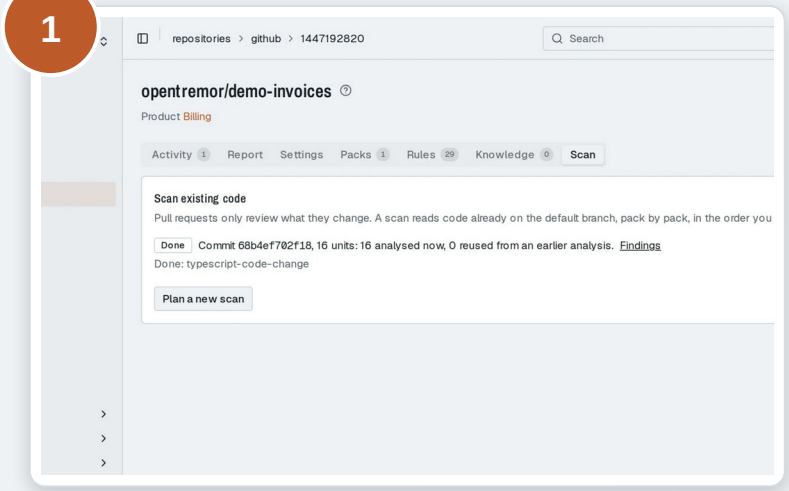
What to notice

- A scan never blocks a pull request. It only shows the current state of the code.
- OpenTremor never sends the same code to the AI model twice.



Find problems already in your code

1



A platform engineer scans the code already on the default branch.

The repository's Scan tab shows what a scan will check. One click scans the main branch.

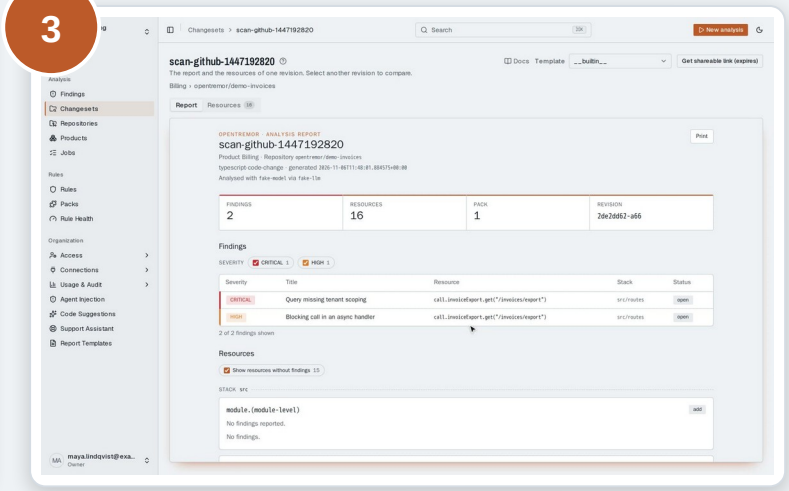
2



OpenTremor finds what the existing code breaks, rule by rule.

The scan finds an old CSV export that returns the invoices of all customers. Each finding names the rule and the risk.

3



The scan's report sorts the findings by severity: a backlog to work through.

The scan report counts the findings by severity: one critical, one high. The team can plan the work and assign it.

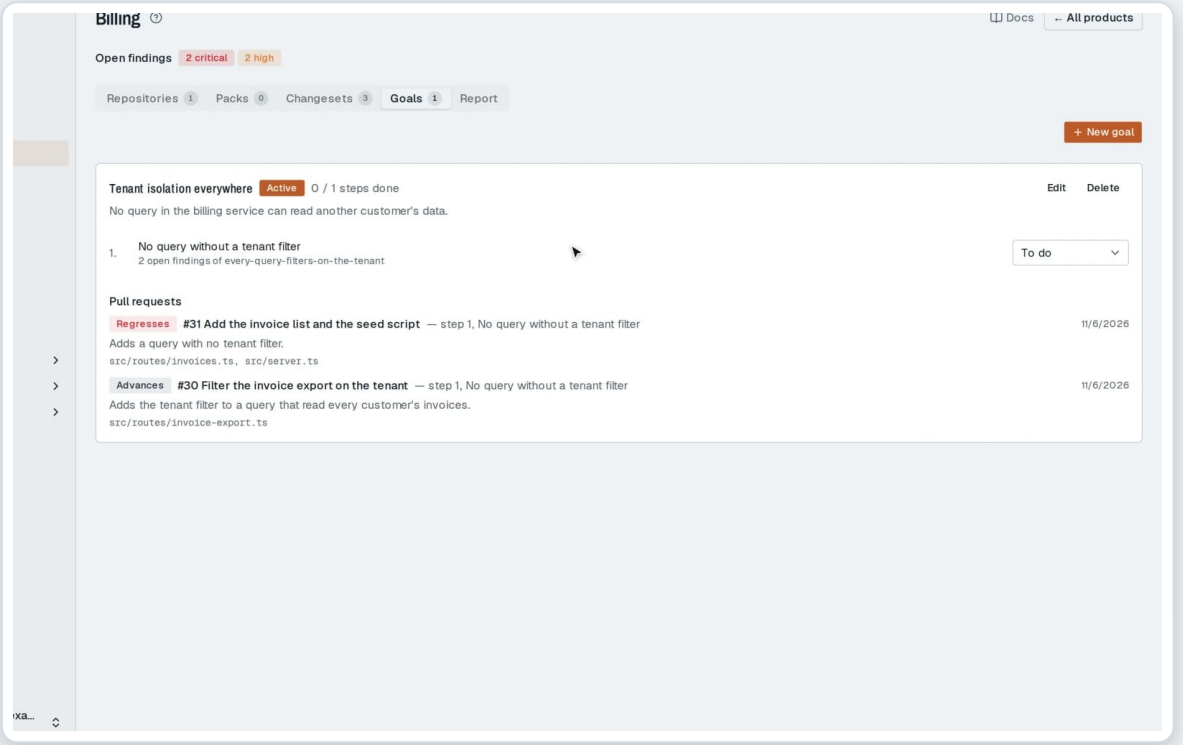
Track an engineering goal across pull requests

“As an engineering director, I want to know if our changes move us toward our goals, not only if they fix single issues.”

Goals like customer data isolation or a migration live in a slide deck. Nobody knows the real progress, and nobody sees when a pull request makes it worse.

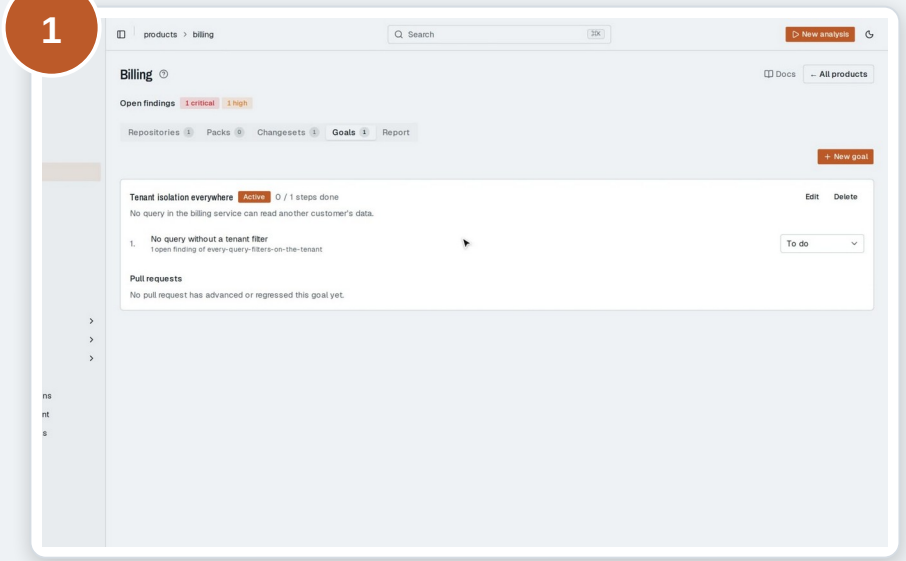
What to notice

- A measured step is done when its rules have no open findings left in the product.
- One pull request never completes the goal by itself. A step is done when its count reaches zero, or when a person marks it as done.



Track an engineering goal across pull requests

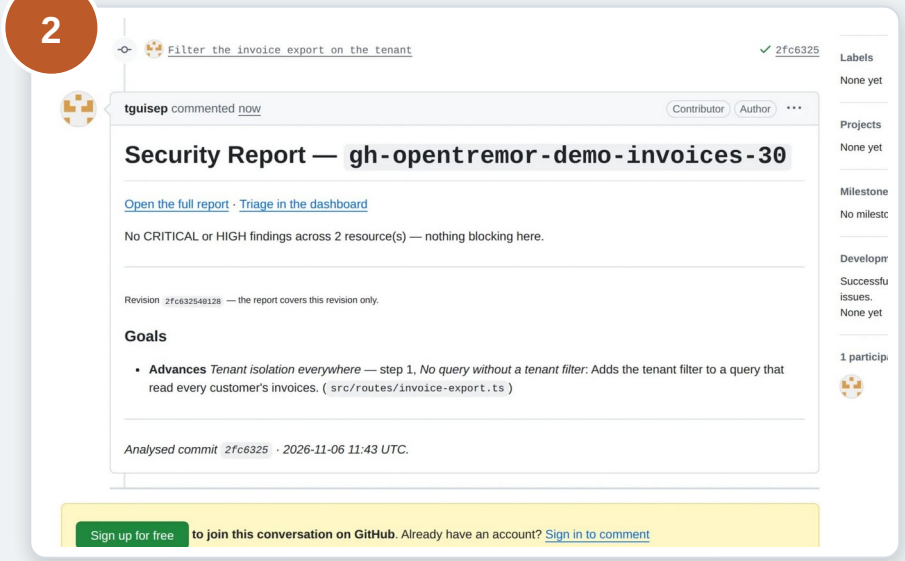
1



An engineering director sets a goal, with a step OpenTremor can measure.

The goal "Tenant isolation everywhere" has a step that OpenTremor measures: the number of open findings for one rule. A scan found one.

2

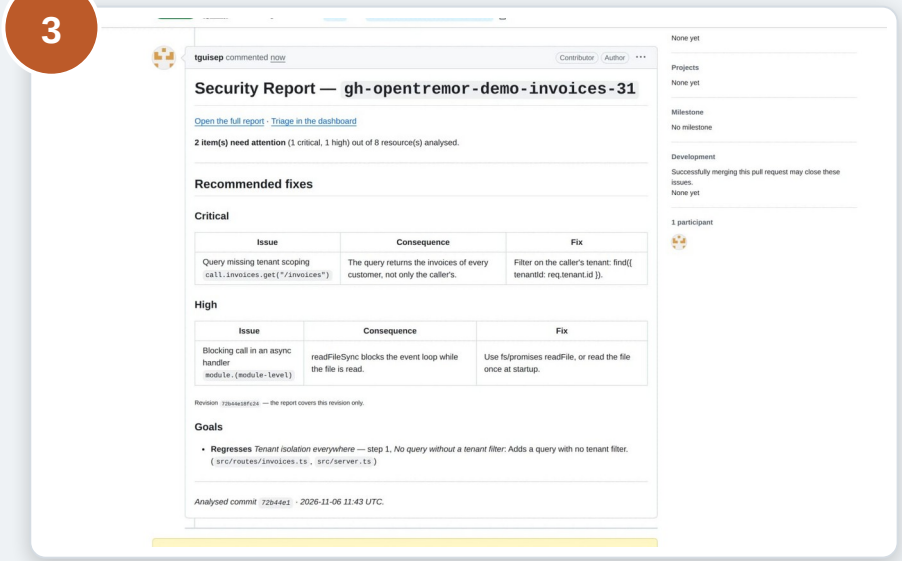


A pull request that fixes old code advances the goal.

A pull request fixes the old export. OpenTremor's comment says that it moves the goal forward, and explains why.

Track an engineering goal across pull requests

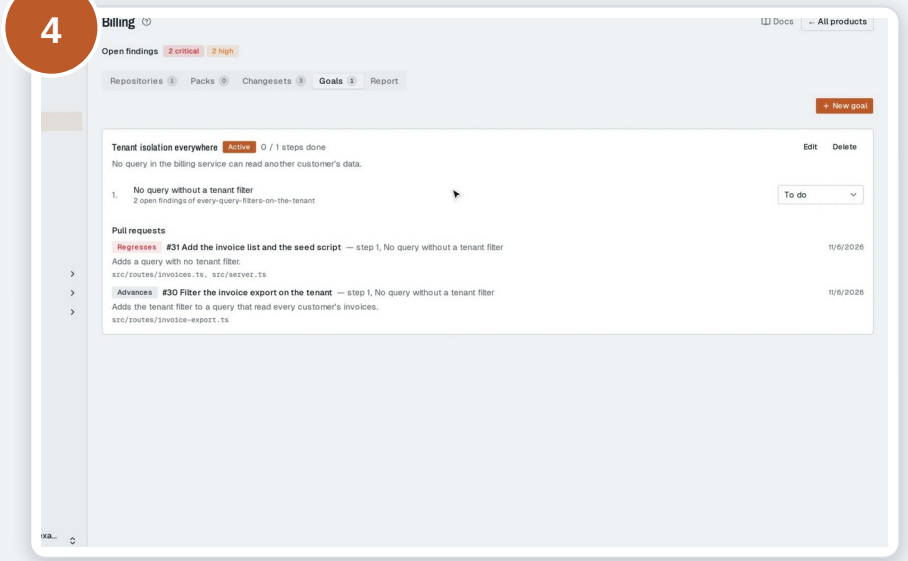
3



One that adds an unfiltered query sets it back, and the comment says so.

Another pull request adds a query that reads every customer's data. The comment says that it moves the goal backward.

4



The goal keeps the record: which pull request moved it, and why.

The goal keeps both pull requests, with the reason and the files, for the next review meeting.

NEXT

Read more

Packs

What OpenTremor reads: languages, IaC, CI and more.

opentremor.dev/packs

Invite your team

Step by step, with screenshots.

docs.opentremor.dev/guide/invite/

Set up your organization

Members, code hosts, rules and the model, in order.

docs.opentremor.dev/personas/org-admin/

See it in motion



opentremor.dev/tour/for/tech-lead · docs.opentremor.dev