

No name, no score, no fix

AI can now read code, yours and the libraries you depend on, and find flaws nobody has written down: **novel vulnerabilities**. They show up with nothing attached, so **your own teams' answers have to carry the call**.

WHAT A FINDING ARRIVES WITH		
	KNOWN	NEW
A public ID	✓	?
A severity score	✓	?
A fix to apply	✓	?
Attack reports	✓	?

What goes quiet

1

Has it been fixed?
Not yet, by definition.

3

Is anyone trying to use it?
No attack data for a flaw nobody has published.

The outside world does some of the work on a known flaw: the vendor, the score, the attack reports. **On a new one it goes silent.**

What carries the call

and who answers it

2

Is something already blocking the way in?
Platform and network teams

3

Can anyone reach it?
App and platform teams

4

How sensitive is it?
The business

5

Can we change it safely, and how fast?
App teams and operations

6-7

Retire it instead? Who would do the work?
The business, and the org chart, sort of

SAME FLAW?

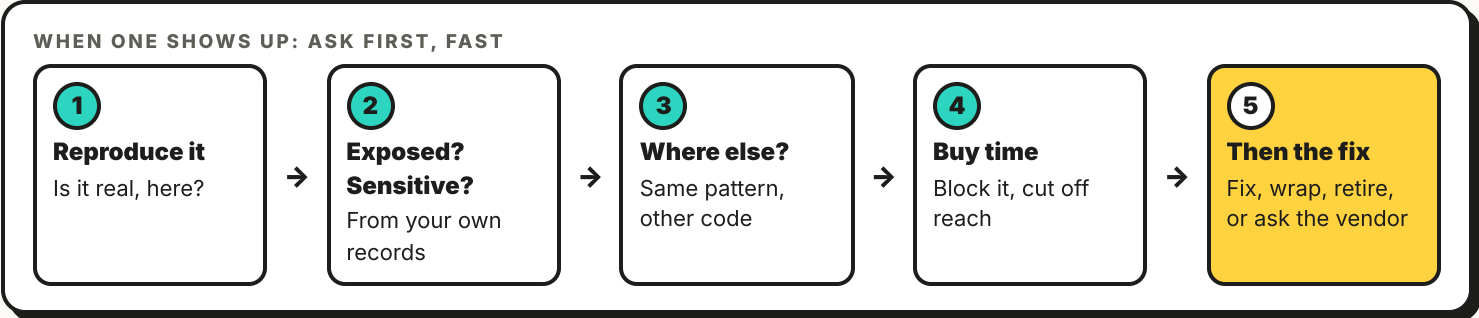
No name to match on
Two findings, two names, no shared ID. Code gets copied and libraries get shared, so a new flaw is rarely in one place. Find the shared piece before anyone files 300 tickets.

CHECK IT

A lead, not a verdict
AI can be wrong, and confidently so. Before changing anything in production, reproduce it: real, in your version, the way you run it. Usually that's the app team.

COSTS MORE

Nobody wrote the fix yet
No patch to apply. Someone has to understand it, size it, and write or request a fix. So **qualify with prejudice**: often the answer is to address it, not rewrite it.



Keep the details need-to-know. An unfixed flaw on a sensitive system is a map. Share the answers widely and the details narrowly until it's fixed. (That's part 3.)

A new source, the same backlog. Known flaws and planned changes already share one change process. Organizations using one AI model reported 10,000+ high or critical findings in about six weeks (its maker's count).