

KONVU VS SEMGREP

Semgrep finds the issues. Konvu proves which ones matter.

Keep Semgrep for detection, custom rules, merge blocking and secrets validation. Supply Chain reachability covers direct dependencies only, and conditionally reachable findings require manual review. Add Konvu for the backlog: it checks environment, configuration and attacker input per finding, reproduces the exploit and ships the fix.

Konvu is not a scanner. It works on the findings Semgrep already produces.

AT A GLANCE

	SEMGREP	KONVU
Question it answers	Does code match a rule? Is a vulnerable dependency installed and its code pattern used?	Can an attacker use this finding in this deployment, and what is the evidence?
Coverage	Reachability on direct deps in 13 language entries, Swift included. None on transitive deps.	Java/Scala/Kotlin, Python, Ruby, JS/TS, C#/NET, Go, PHP, Rust; builds its own dependency graph
What it checks	Vulnerable version and code pattern both match. Conditionally reachable findings require manual review.	Installed, environment, reachable, attacker-controllable, each met or not met with code evidence
Verdict	Reachable, unreachable, conditionally reachable	Exploitable, False Positive, Needs input or Inconclusive
Fixes	Autofix PRs: Code (beta), Supply Chain in JS/Python	Draft fix PRs or an agent plan; exploit re-run on the fix
Write-back	Triage state inside Semgrep	Supply Chain dismissals set Semgrep triage state to Ignored. Jira tickets on Enterprise.
Pricing	Free up to 10 contributors; Teams: Code \$30, Supply Chain \$30 per contributor/month	\$2,000/month for 400 confident verdicts; inconclusive runs are free

Where Semgrep is stronger

- Custom YAML rules, and merge blocking that fails CI.
- Multimodal triage that learns from your ignore comments.
- Malware blocking at the PR and local secrets validation.
- Direct-dependency reachability in 13 language entries, Swift included, and a free edition for 10 contributors.

Where Konvu fits

- Transitive deps: Konvu builds its own dependency graph.
- Conditionally reachable findings: Konvu checks OS, runtime and configuration against your code.
- Supply Chain findings; Multimodal primarily supports Code.
- The exploit reproduced against a copy of your app.

RUNNING BOTH

Semgrep stays in CI. Konvu works the backlog.

01

Connect Semgrep

Add a Web API token. Konvu pulls Code and Supply Chain findings daily and on demand.

02

Konvu decides

A verdict and evidence per finding. Exploitable ones reproduced and fixed.

03

Dismissals go back

Supply Chain dismissals become Ignored in Semgrep. Code verdicts stay in Konvu.

Detection is solved. The decision is not.

Full comparison with every source: konvu.com/compare/konvu-vs-semgrep

Konvu docs: docs.konvu.com

Sources checked October 13, 2026: semgrep.dev/pricing, docs.semgrep.dev (supported languages, Supply Chain overview, Multimodal, autofix, blocking), semgrep.dev/blog (Multimodal), docs.konvu.com, konvu.com/pricing, konvu.com/integrations/semgrep.