

KONVU VS ENDOR LABS

Endor Labs traces the call path. Konvu checks exploitability.

Keep Endor Labs for detection, SBOM and VEX export, and backported patches. Add Konvu for reachable backlogs too big to read: it checks environment, attacker input and configuration, proves the exploit, ships the fix. Endor's reachability traces direct and transitive dependencies in eight languages; Konvu also assesses PHP and Rust.

Konvu is not a scanner. It reads Endor findings; dismissals stay in Endor.

AT A GLANCE

	ENDOR LABS	KONVU
Question it answers	Is there a path from my code to the vulnerable function?	Can an attacker use this finding in your application and environment, and what is the evidence?
Reachability languages	Function-level: Java, Python, JS/TS, Go, .NET, Kotlin, Scala, Ruby. None listed for PHP, Rust	Java/Scala/Kotlin, Python, Ruby, JS/TS, C#/.NET, Go, PHP, Rust. No C/C++ or Swift
What it checks	Call path through direct and transitive dependencies; KEV, fix availability, upgrade risk	Installed, environment, reachable, attacker-controllable; then the exploit is reproduced
Verdict	Reachable, Unreachable or Potentially Reachable	Exploitable, False Positive, Needs input or Inconclusive
Fixes	Version-bump PRs, AURI Agents, Endor Patches	Tested fix PR or agent plan; exploit re-run on the fix
Write-back	Jira tickets; exports to GitHub Advanced Security, Wiz and S3	Read-only on Endor. Jira (Enterprise); dismissals to Snyk, Semgrep, Dependabot, GitLab, ArmorCode
Pricing	Per contributing developer, no list prices, free tier	\$2,000/mo, 400 confident verdicts; inconclusive runs free

Where Endor Labs is stronger

- A scanner with its own vulnerability database.
- Call-graph reachability into direct and transitive dependencies in eight languages, plus a no-build fallback.
- Upgrade impact analysis and backported patches.
- SBOM and VEX export, cloud Bitbucket and Azure DevOps, and a free tier.

Where Konvu fits

- Reachable backlogs too big to read each call path by hand.
- Reachable findings that are safe: input is controlled, the dangerous option is off, or a safe API is used.
- PHP and Rust, listed without reachability in Endor's SCA table.
- Exploit proof against a copy of your app, and a recorded reason for every dismissal.

RUNNING BOTH

Endor keeps the record. Konvu holds the verdict.

01

Connect Endor Labs

A read-only API key, one namespace. Konvu reads SCA findings and reachability tags daily and on demand.

02

Konvu decides

Uses Endor's reachability with its own analysis, then checks environment and attacker controllability.

03

Endor stays the record

Your team records the outcome in Endor with Konvu's evidence. Exploitable findings get a tested fix PR.

Detection is solved. The decision is not.

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

Konvu docs: docs.konvu.com

Sources checked October 13, 2026: docs.endorlabs.com (reachability, call graphs, SCA, findings, upgrade impact, Endor Patches, SBOM/VEX, integrations, Outpost), endorlabs.com (pricing, agentic workflows, customers), docs.konvu.com, konvu.com/pricing, konvu.com/integrations/endor-labs.