# SCA vs SAST

What each tool actually does (and doesn't) | March 2026

konvu

konvu.com/compare/sca-vs-sast

**TL;DR** SCA and SAST are not competing approaches. SCA finds known vulnerabilities in third-party dependencies (77-90% of your codebase). SAST finds vulnerabilities in first-party code. You need both. Start with SCA (faster to deploy, fewer false positives), then add SAST within a quarter.

## Head-to-head overview

| Category | SCA | SAST |
|---|---|---|
| **What it scans** | Third-party dependencies (lockfiles, manifests, container images) | First-party source code your team wrote |
| **What it finds** | Known CVEs in libraries, license risks, outdated components | SQL injection, XSS, hardcoded credentials, insecure patterns |
| **False positive rate** | 2-10% (technically accurate but 80-98% may be unreachable without reachability analysis) | 15-60% in real-world deployments (98% unexploitable at runtime per StackHawk) |
| **Key technique** | Manifest/lockfile parsing + vulnerability database matching + reachability analysis | AST pattern matching + taint analysis (intra- and inter-procedural) |
| **Deployment speed** | Minutes: scans manifests and lockfiles, no code compilation needed | Hours to days: requires rule tuning, baseline triage, CI integration |
| **Output format** | CVE list with severity, affected package, fix version, SBOM (CycloneDX/SPDX) | Vulnerability findings with CWE, code location, data flow trace, remediation guidance |

## What the data shows

| Metric | Source | SCA | SAST |
|---|---|---|---|
| **False positive rate** | Industry benchmarks | 2-10% (findings technically correct) | 15-60% in production (vendor claims: <1-12%) |
| **Actionable findings** | Reachability vendors | 2-20% reachable (80-98% noise without reachability) | ~39% detection rate (EASE 2024, best single tool) |
| **Attack surface covered** | Synopsys OSSRA 2024 | 77-90% of codebase (third-party) | 10-23% of codebase (first-party) |
| **Remediation path** | Practitioner consensus | Update dependency version (clear, sometimes breaking) | Fix code pattern (requires developer effort, context-dependent) |
| **Annual alert volume** | Industry average | Avg 569K total alerts/yr across tools, 202 need immediate action | 17 new vulns/month per app, teams fix 6 (debt accumulates 3x) |

## Pricing at a glance

| Company size | SCA (estimated) | SAST (estimated) |
|---|---|---|
| **Startup (<20 devs)** | $0 free stack: Trivy + Dependabot + OWASP Dependency-Check | $0 free stack: Semgrep OSS + CodeQL (public repos) |
| **Mid-market (20-200 devs)** | $15K-$70K/yr (Snyk Team, Semgrep Supply Chain, or Endor Labs) | $15K-$70K/yr (Semgrep Teams, SonarQube Enterprise, or Checkmarx One) |
| **Enterprise (200+ devs)** | $70K-$300K+/yr (Snyk Enterprise, Black Duck, or Endor Labs with reachability) | $70K-$300K+/yr (Checkmarx, Fortify, Veracode, or full platform bundles) |

## When to pick which

### Start with SCA when:

- Largest attack surface: 77-90% of your code is third-party dependencies
- Fewer false positives (2-10% vs 15-60% for SAST), clearer remediation path
- Faster deployment: manifest/lockfile scanning in minutes, not days
- 70% of critical security debt comes from third-party code (Veracode 2025)

### Start with SAST when:

- Custom code is the primary risk (auth, data handling, business logic)
- Regulatory requirements demand first-party code analysis
- Need to catch injection, XSS, hardcoded credentials in your own code
- Already have SCA coverage and need to complement it

Sources: OWASP, Synopsys OSSRA, Endor Labs Station 9, Veracode State of Software Security. Full article: **konvu.com/compare/sca-vs-sast**

The bottleneck is rarely detection. It's triage. If triage is still the problem, that's what Konvu solves. **konvu.com/demo**