

Best AI Code Review and Security Scanning Tools (2026)

Seven AI code review and security scanning tools compared on what they do, what they cost and who should buy them, using vendor pricing as of October 2026.

Introduction

The tools that review your code now fall into two groups, and they have been growing toward each other. One group is LLM reviewers. They read a pull request, leave comments in plain English and suggest fixes. The other group is security scanners. They run static analysis (SAST), dependency checks (SCA) and secret detection against rules, and these days they bolt on AI to explain or patch what they find. Most teams need some of each. An LLM reviewer catches logic and design problems a rule engine never will, and a rule engine flags a known injection pattern every time.

This roundup covers seven products: four AI-first reviewers and three security platforms. Pricing comes from each vendor's own pricing page as of October 2026. scored.tools rates tools on five criteria: quality of output, ease of use, pricing value, API and integration quality, and how well the tool solves the problem it claims to solve. None of these seven has a published scored.tools rating yet, so none gets a score below. Entries are listed alphabetically, and the order says nothing about rank.

1. CodeRabbit

Score: Not yet rated

Coderabbit is an AI reviewer that comments on pull requests and also runs in the CLI and the IDE. Its pricing page lists learnings (it picks up your team's feedback over time), chat in the PR, one-click fixes, pre-merge checks and what it calls slop detection on every paid plan. Team adds unit test generation and Slack triage. The security features sit on the Advanced tier: security-informed prioritization, continuous security monitoring, PR security reviews and blast-radius analysis. The limits are worth reading before you buy. Essentials allows 1 linked repo and 5 reviews per developer per hour. Team raises that to 5 repos and 8 reviews an hour. A monorepo shop will be fine on the cheaper plans, but a team spread across dozens of repos will hit the cap fast.

Best for: Teams that want an LLM reviewer in GitHub or GitLab PRs and work in a small number of repos.

Pricing: Essentials \$30/user/month (\$24 billed annually), Team \$60 (\$48 annually), Advanced \$72/user/month billed annually, Enterprise by quote. Deep Scan and CodeRabbit Engineer are sold separately.

2. GitHub Copilot code review and GitHub Advanced Security

Score: Not yet rated

GitHub Copilot code review works on every paid Copilot plan (Pro, Pro+, Max, Business and Enterprise), and GitHub's docs say the Free plan does not include it. It now bills in AI credits. GitHub estimates a review costs about \$0.05 to \$1 at Lite effort and \$0.25 to \$5 at Balanced effort, plus Actions minutes, and automatic reviews are charged to the PR author. That spread is wide, so set a budget. The security side is sold separately per active committer. Secret Protection (\$19) covers push protection, validity checks and Copilot-based detection of unstructured secrets like passwords. Code Security (\$30) adds CodeQL analysis, Dependabot, and Copilot Autofix for JavaScript, TypeScript, Java and Python alerts. The main reason to buy is that everything lives inside GitHub. If you host on GitLab or Bitbucket, it is not an option.

Best for: Organizations already on GitHub Enterprise or Copilot Business that want review and scanning without another vendor.

Pricing: Code review uses AI credits on paid Copilot plans. Secret Protection \$19 and Code Security \$30 per active committer per month.

3. Greptile

Score: Not yet rated

Greptile is an AI reviewer that indexes the whole codebase so that its PR comments can account for code outside the diff. Its pricing is simpler than most. Pro is a flat \$30 per seat with unlimited repositories and users, 50 credits per seat, custom rules and unlimited external app connections. Extra credits cost \$1 each. Enterprise adds self-hosting, SSO/SAML and GitHub Enterprise support, which matters if your security team won't send source code to a third-party cloud. Greptile does not market itself as a SAST or dependency scanner, so plan to pair it with one of the security tools below.

Best for: Teams with large, interconnected codebases that want review comments grounded in the full repo, and teams that need a self-hosted reviewer.

Pricing: Free Starter (1 developer, 50 credits/month), Pro \$30/seat/month with a 14-day trial, Enterprise by quote. Free for qualifying open source projects, 50% off for early-stage startups.

4. Qodo

Score: Not yet rated

Qodo (formerly CodiumAI) sells agentic PR review, a rules system with no limit on rules, pre-PR review skills in the IDE and cross-repo context. Pro Team is metered in pooled credits at \$0.012 each. Qodo's page says 2,500 credits buys about 18 reviews a month, 5,000 about 36, and 20,000 about 144. That works out to roughly \$1.67 per review at list rates. You can set a spending cap, and credits expire at the end of each cycle. There is no permanent free tier, only a 14-day trial, and the page heads the plan "unlimited" while also describing credit metering, so read the terms carefully. SSO, audit logs and on-prem deployment are Enterprise only.

Best for: Teams under 30 developers who want per-review billing and IDE-side review before a PR is opened.

Pricing: 14-day free trial, Pro Team from \$30/month (2,500 credits), Enterprise by quote for 30+ users.

5. Semgrep

Score: Not yet rated

Semgrep is a rule-based static analyzer with an AI layer, which its pricing page calls Semgrep Multimodal, for triage and fix suggestions metered in AI credits. You buy it by product. Code (SAST) and Supply Chain (SCA) are \$30 per contributor each on Teams, and Secrets is \$15. The free edition is generous. Up to 10 contributors and 10 private repos get Code and Supply Chain, including Pro rules and cross-file analysis, at no cost. Semgrep counts a contributor as anyone who committed to a scanned private repo in the last 90 days, so contractors and occasional committers add to the bill. Writing custom rules is the strongest reason to pick it over a closed scanner.

Best for: Security-minded engineering teams that want to write their own rules and run scans in CI.

Pricing: Free for up to 10 contributors and 10 private repos. Teams from \$30/contributor/month per product (Secrets \$15). Enterprise by quote.

6. Snyk

Score: Not yet rated

Snyk covers more ground than anything else here: open source dependencies, SAST (Snyk Code), infrastructure as code and containers, with IDE, CLI and SCM integrations on every plan. The free tier allows 100 Snyk Code tests a month across 5 projects, which is enough to try it and too little to run on every commit. Team starts at \$25/month for up to 10 developers and 1,000 code tests. Enterprise moves to prepaid credits at \$1 each. Snyk Code and Open Source cost 1 credit per active contributor per day, so budgeting takes some arithmetic. The newer Evo agentic features, including coding agent security and AI pentesting, are Enterprise only.

Best for: Teams that want one vendor for dependencies, containers, IaC and first-party code.

Pricing: Free (100 code tests/month, 5 projects), Team from \$25/month for up to 10 developers, Enterprise on credits at \$1 per credit.

7. SonarQube

Score: Not yet rated

Sonarqube has been the default code quality gate for a long time, and its cloud product now bundles Advanced Security into the Team plan. Pricing is by lines of code. Only private projects count, and only the largest branch counts. The free tier covers private projects up to 50k LOC, which suits a small product. Sonar's pricing page lists Team from \$34/month for 100k LOC, while its own FAQ cites \$32, so confirm the price at checkout. On the AI side, Sonar now lists Gatar AI Code Review at \$20/user/month billed annually, and a Remediation Agent on Enterprise. Sonar's strength is rule coverage across 45+ languages on Enterprise, along with OWASP and MISRA reporting for teams with compliance requirements.

Best for: Teams that need quality gates and compliance reporting more than conversational PR review.

Pricing: Free up to 50k private LOC, Team from about \$34/month for 100k LOC, Enterprise by quote for 50+ developers. Gatar AI review \$20/user/month annually.

Comparison table

Tool	Type	Free tier	Entry paid price	Billing unit	Score
Coderabbit	AI PR review (security on Advanced)	No (trial only)	\$24/user/mo annual	Per user	Not yet rated
Github Copilot	AI review + CodeQL, secrets	No code review on Copilot Free	\$19/committer (Secret Protection)	AI credits / active committer	Not yet rated
Greptile	AI PR review, full-repo context	Yes, 1 developer	\$30/seat/mo	Per seat + credits	Not yet rated
Qodo	Agentic PR + IDE review	14-day trial	\$30/mo (2,500 credits)	Pooled credits	Not yet rated
Semgrep	SAST, SCA, secrets + AI triage	Yes, 10 contributors	\$15/contributor (Secrets)	Per contributor per product	Not yet rated

Tool	Type	Free tier	Entry paid price	Billing unit	Score
Snyk	SCA, SAST, IaC, containers	Yes, 100 code tests/mo	From \$25/mo	Tests / credits	Not yet rated
Sonarqube	Quality gates + SAST	Yes, 50k LOC	~\$34/mo (100k LOC)	Lines of code	Not yet rated

Final picks

Without scored.tools ratings for these seven, these picks follow from pricing and documented features alone.

- **Cheapest serious security baseline:** Semgrep's free edition, with Code and Supply Chain for up to 10 contributors, is hard to beat for a small team.
- **Broadest scanner coverage:** Snyk, if you need containers and IaC alongside code and dependencies.
- **Already on GitHub Enterprise:** Github Copilot code review plus Code Security keeps everything in one bill and one UI. Watch the per-review AI credit costs.
- **AI reviewer with flat pricing:** Greptile Pro at \$30 per seat with unlimited repos avoids the per-repo caps on Coderabbit's lower tiers.
- **AI reviewer for a few focused repos:** Coderabbit Team, for its learnings and test generation.
- **Compliance-heavy teams:** Sonarqube for quality gates and OWASP/MISRA reporting.

A sensible setup in 2026 pairs one AI reviewer with one rule-based scanner. The reviewer catches what the rules can't express, and the scanner catches what a model might miss.