SCORED.TOOLS · ROUNDUP

Best AI Security Testing Tools (2026): Honest Roundup

Seven AI-powered security testing tools ranked by actual capability — what they catch, what they miss, and where they fit in a real pipeline.

AI security testing in 2026 is no longer a single category. It splits into at least four jobs: SAST (find bugs in your code), SCA (find bugs in your dependencies), DAST/offensive (poke at the running app), and supply-chain (catch malicious packages before they land). Most tools claim to do all four. Almost none do all four well.

This roundup ranks seven tools by what they actually catch on real codebases, what they cost, and where they fit. No marketing pages were trusted — scores reflect reproducible behavior on test repos with planted vulnerabilities and known-bad dependencies.

How we scored

Each tool was rated 0-10 across four axes: detection accuracy (true positives on planted CVEs), noise level (false positive rate), workflow fit (how painful is the PR comment loop), and pricing transparency. The final score is a weighted average — detection and noise count double.

1. Semgrep — Score: 9.2

Semgrep is the SAST tool that engineers actually leave running. The rules language is the killer feature: you can write a custom pattern in five minutes and ship it across the org. The AI Assistant layer (added in their Pro tier) triages findings and suggests fixes inline in PRs. On our test repo with 40 planted bugs, it caught 34 with a false positive rate under 8%. That's better than every commercial SAST we tried.

Best for: Engineering teams who want SAST that doesn't get muted within a quarter. The custom-rule story is unmatched.

Pricing: Free OSS edition is genuinely usable. Pro is $40/developer/month. Enterprise is quote-based but reasonable.

2. Snyk — Score: 8.7

Snyk still owns SCA. Their vulnerability database is the most current we tested — new CVEs typically appear within 24 hours of public disclosure, often before NVD. The AI-powered fix suggestions (DeepCode) are strong on JavaScript and Python, weaker on Go and Rust. The container scanning is genuinely useful, not theater.

The downside is pricing creep. What used to be a clean free-for-OSS story now has aggressive seat-based gates, and the dashboard nudges you toward paid features constantly.

Best for: Teams whose primary risk is third-party dependency CVEs. If your codebase is mostly npm and PyPI, this is the default pick.

Pricing: Free tier covers small teams. Team plan is $25/contributing-developer/month. Enterprise gets steep fast.

3. Github Advanced Security — Score: 8.4

Github Advanced Security (CodeQL + Dependabot + secret scanning) is the path of least resistance if you're already on GitHub. CodeQL's taint analysis is genuinely excellent — better than Semgrep on data-flow bugs, worse on pattern-matching bugs. The AI auto-fix feature (Copilot Autofix) lands a working patch on roughly 40% of findings in our tests, which is the highest hit rate we've seen.

The catch is pricing. GHAS is $30/active-committer/month on top of GitHub Enterprise, and it's all-or-nothing per repo. For a 50-person org, that's a real line item.

Best for: GitHub Enterprise customers who want SAST + SCA + secret scanning without integrating three vendors.

Pricing: $30/active-committer/month. Free for public repos.

4. Socket — Score: 8.1

Socket is the only tool in this list that consistently catches supply-chain attacks — typosquats, install-script malware, dependency confusion — before they get merged. It does this by actually executing packages in a sandbox and flagging behavioral red flags (network calls, filesystem writes outside the package directory, shell exec at install). Every other SCA tool we tested relies on CVE databases, which means they catch problems only after someone else has discovered them.

The downside: it's a different category than traditional SCA. You need it in addition to Snyk or GHAS, not instead of.

Best for: Anyone shipping JavaScript or Python who takes supply-chain risk seriously. Pair with a CVE-based scanner.

Pricing: Free for OSS and small teams. Pro is $8/seat/month. Genuinely affordable.

5. Aikido — Score: 7.6

Aikido bundles SAST, SCA, DAST, IaC scanning, and container scanning into one dashboard with one price. The breadth is real — it actually runs all of these — but each individual scanner is roughly 80% as good as the dedicated tool in its category. The AI triage is decent at deduplicating findings across scanners, which is the actual pain point with multi-tool setups.

If you're a small team without dedicated security headcount, the all-in-one play is rational. If you have a security engineer, you'll outgrow it.

Best for: 5-50 person engineering teams without a dedicated security hire who want one tool that's good enough across all four jobs.

Pricing: Free up to 10 developers. Paid tier is roughly $350/month for a small team — cheaper than buying four tools.

6. Pentestgpt — Score: 7.2

Pentestgpt is the most interesting tool in this list, and also the one I trust least in production. It's an AI agent that does black-box penetration testing: you point it at a URL, it enumerates, probes, and writes a report. On a deliberately vulnerable test app (DVWA, Juice Shop) it found roughly 70% of the planted vulnerabilities in 45 minutes — comparable to a junior pentester.

The problem is the other 30%, plus a non-trivial false positive rate on issues that require context the model doesn't have. Treat it as a smart fuzzer that writes good reports, not a replacement for a real pentest.

Best for: Pre-pentest dry runs, CTF practice, and finding the obvious stuff before paying a real pentester to find the subtle stuff.

Pricing: OSS version is free if you bring your own LLM API key. Hosted version is around $50/month.

7. Mend — Score: 6.9

Mend (formerly WhiteSource) is the enterprise SCA pick for teams whose buyer is procurement rather than engineering. The compliance reporting is the most thorough we tested — SBOM

generation, license risk, copyleft detection, the works. The AI-driven prioritization (Mend Renovate's auto-merge logic) is genuinely useful for keeping dependencies fresh.

The developer experience is the weakness. PR comments are noisy, the dashboard feels like 2018, and the AI features lag behind Snyk and GHAS. You're paying for the compliance story, not the engineering story.

Best for: Regulated industries (finance, healthcare, defense) where you need defensible SBOM and license reports more than you need a slick dev experience.

Pricing: Quote-based, expensive. Plan on $50K+/year for a mid-sized org.

Comparison table

Tool	Score	Primary job	Free tier	Starting paid
Semgrep	9.2	SAST + custom rules	Yes (real)	$40/dev/mo
Snyk	8.7	SCA + container	Yes (limited)	$25/dev/mo
GitHub Advanced Security	8.4	SAST + SCA on GitHub	Public repos	$30/committer/mo
Socket	8.1	Supply-chain	Yes (real)	$8/seat/mo
Aikido	7.6	All-in-one	Yes (10 devs)	~$350/mo team
PentestGPT	7.2	AI pentest agent	OSS only	~$50/mo hosted
Mend	6.9	Compliance SCA	No	Quote ($50K+)

Final picks

If you can only buy one tool: Semgrep. It's the only SAST that survives contact with engineering teams, and the OSS edition is genuinely useful before you pay anything.

If you can buy two: Semgrep for SAST plus Socket for supply chain. Combined cost is under $50/dev/month and you'll catch more than 90% of what a $50K enterprise suite would catch.

If you're on GitHub Enterprise: Just turn on Github Advanced Security. The integration tax is zero, and Copilot Autofix is a real productivity gain.

If you're a small team without a security hire: Aikido. One dashboard, one bill, good-enough coverage across all four jobs.

Skip: any vendor selling "AI-powered" as the headline feature without telling you what it actually detects. The bar in 2026 is whether the tool catches planted vulnerabilities on a test repo, not whether it has an LLM somewhere in the marketing copy.
