Phoenix Security Launches Phoenix Purple and a New Agentic SDLC Security Strategy: Close the Tap, Burn the Backlog

Phoenix Security Purple Agentic SDLC Security

LONDON, UNITED KINGDOM, June 29, 2026 /EINPresswire.com/ -- Phoenix Security today launched Phoenix Purple, graph-native AI scanning for AI-generated code, alongside a new strategy for securing the software development lifecycle now that AI agents write a large share of new code and open pull requests by the minute. The strategy fits in two moves: close the tap on new vulnerabilities entering the code, and burn down the backlog already sitting in the estate. Underneath sit three pillars of control — control the agent, close the tap, burn the backlog — and a single knowledge graph that runs under all of them.

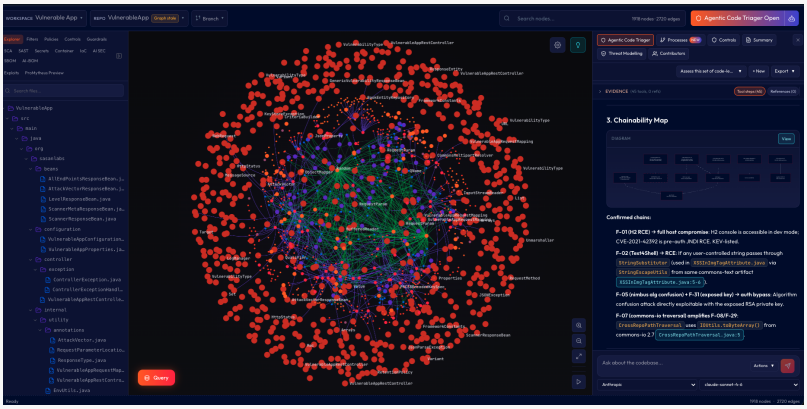
Detection was never the bottleneck. Teams have always had more findings than they could act on. What changed is volume and speed: agents generate code faster than any review process built for human-paced development can keep up, and they pull in packages no one reads. Adding another scanner only makes the list longer. Phoenix attacks the problem from both ends — stopping new vulnerabilities at the source while grinding down the existing pile — with the knowledge graph as the connective layer that makes both halves one program rather than two tools bought separately.

> This is the first security program I can take to a CTO and have them come out ahead on cost, not behind."
>
> *Francesco Cipollone*

What's new with Phoenix Purple. Check the AI Cost calculator.

Phoenix Purple scans AI-generated code inside the agent's working session, on the graph, in seconds. Because the

graph tells the model where to look — which functions are reachable, which paths carry untrusted input, which parts of the repo a change actually touches — a smaller model does the work that would otherwise require running a frontier model across the entire codebase on every change.



Phoenix Security Purple Agentic SDLC

That design choice is what changes the economics. On a modeled fleet of 26 repositories at 250,000 lines each, scanned monthly, a file-by-file harness running on a frontier model came in at roughly $598 per month and $64.30 per confirmed vulnerability.

Phoenix Purple on the same model and fleet ran at $74.88 per month and $3.60 per confirmed vulnerability — about eight times less across the fleet, and roughly 87% lower cost for the same findings on the same model. The gap widens as codebases grow, because file-by-file cost climbs with every line added while graph-native cost stays tied to real findings.

The three pillars

Control the agent. Phoenix reads a team's backlog — the best record of what they get wrong — and turns confirmed findings into rules the agent follows every session, written as plain constraints in the files agents already read (.mdc, AGENTS.md, .cursorrules). Classes of bugs a team has already paid to fix stop being regenerated. Each new finding becomes a candidate for a new guardrail, so the program compounds rather than resetting each sprint.

Close the tap. Four controls operate before code reaches the main branch: architectural rules and threat modeling at design, agent-level scanning in session, package intelligence at install that checks what a dependency actually does rather than waiting for a CVE, and a pull-request gate that re-runs the same checks so nothing routes around a single chokepoint.

Burn the backlog. Phoenix Orange aggregates every scanner into one queue, removes what isn't reachable, attributes each finding to its real owner, and ranks what survives by genuine risk — turning a list of, say, 112,000 findings into roughly 300 worth a person's attention. Phoenix Green then reads that ranked queue and writes the fix as a pull request, with threat context in the description. Low-consequence fixes are fast-tracked; breaking or dangerous ones wait for a human. Engineers review fixes instead of hunting for them.

A security program a CTO can come out ahead on

“This is the first security program I can take to a CTO and have them come out ahead on cost, not behind. You were going to pay for AI in the SDLC anyway. Run it on the graph, and the token bill drops while your agents speed up. The same graph that makes scanning cheap is the road your developers and their agents drive on. Security stops being the thing that slows the lane and becomes the thing that lets you open it. Less doom, more action.”

— Francesco Cipollone, CEO & CISO, Phoenix Security

The controls are composable rather than a single bundled, per-scan product: start with aggregation to get a number you trust, add the agent firewall when the developer fleet needs it, add scanning when pull-request velocity is the bottleneck, add remediation when the backlog is the constraint. The number that tells a team it's working is burn rate — findings closed per week should outpace findings created per week.

Availability

Phoenix Purple is available now. Details on graph-native AI scanning for AI-generated code are at phoenix.security/phoenix-purple-ai-sast-sca-ai-generated-code, and teams can model their own costs with the scan cost calculator at ai-scan-cost.phoenix.security. To schedule a demonstration, visit phoenix.security/request-a-demo.

Phil Moroni
Phoenix Security
+1 919-594-8888
email us here
Visit us on social media:
LinkedIn
Instagram
Facebook
YouTube
X
