Fortify Agentic Analyzer

Fortify Agentic Analyzer (FAA) brings agentic analysis into your existing Fortify workflow to find the business-logic and authorization flaws that rules-based analysis cannot reach.



Benefits

- Catches issues that are beyond the reach of SAST
- One taxonomy, one ASPM view, no new findings silo
- Included under your existing OpenText Fortify SAST license
- Bring your own LLM; run it selectively, like a pen test

The hardest flaws to find are the ones that turn on what an application is meant to do: broken authorization, business-logic gaps, multi-step injection. They surface in code of any age or origin, and they lie beyond what any fixed set of rules can express. Finding them takes reasoning about the code, not another rule, and teams need that depth without standing up a second tool or a second triage queue.

A hybrid model: Deterministic SAST plus agentic depth

OpenText™ Fortify™ SAST stays the fast, deterministic first line of defense on every build. Fortify Agentic Analyzer (FAA) adds a deeper, agentic pass you run selectively on higher-risk applications, the way you would schedule a penetration test. FAA drives an AI coding agent over an application codebase to find vulnerabilities beyond source-to-sink dataflow: flawed authentication and authorization design, business-logic weaknesses, and subtle injection and persistence patterns. It is language-agnostic, covering web applications, APIs, microservices, and IaC.

- **Fortify Agentic Analyzer (FAA):** Agentic analyzer that finds flaws beyond the reach of rules-based analysis; included under SAST, bring your own LLM.
- **Fortify SAST:** Deterministic, daily first line of defense; the baseline FAA complements and dedupes against.
- **SSC/Fortify on Demand:** The shared ASPM system of record where agentic findings sit in one taxonomy with SAST, DAST, and SCA.

The proof—it finds real, critical flaws

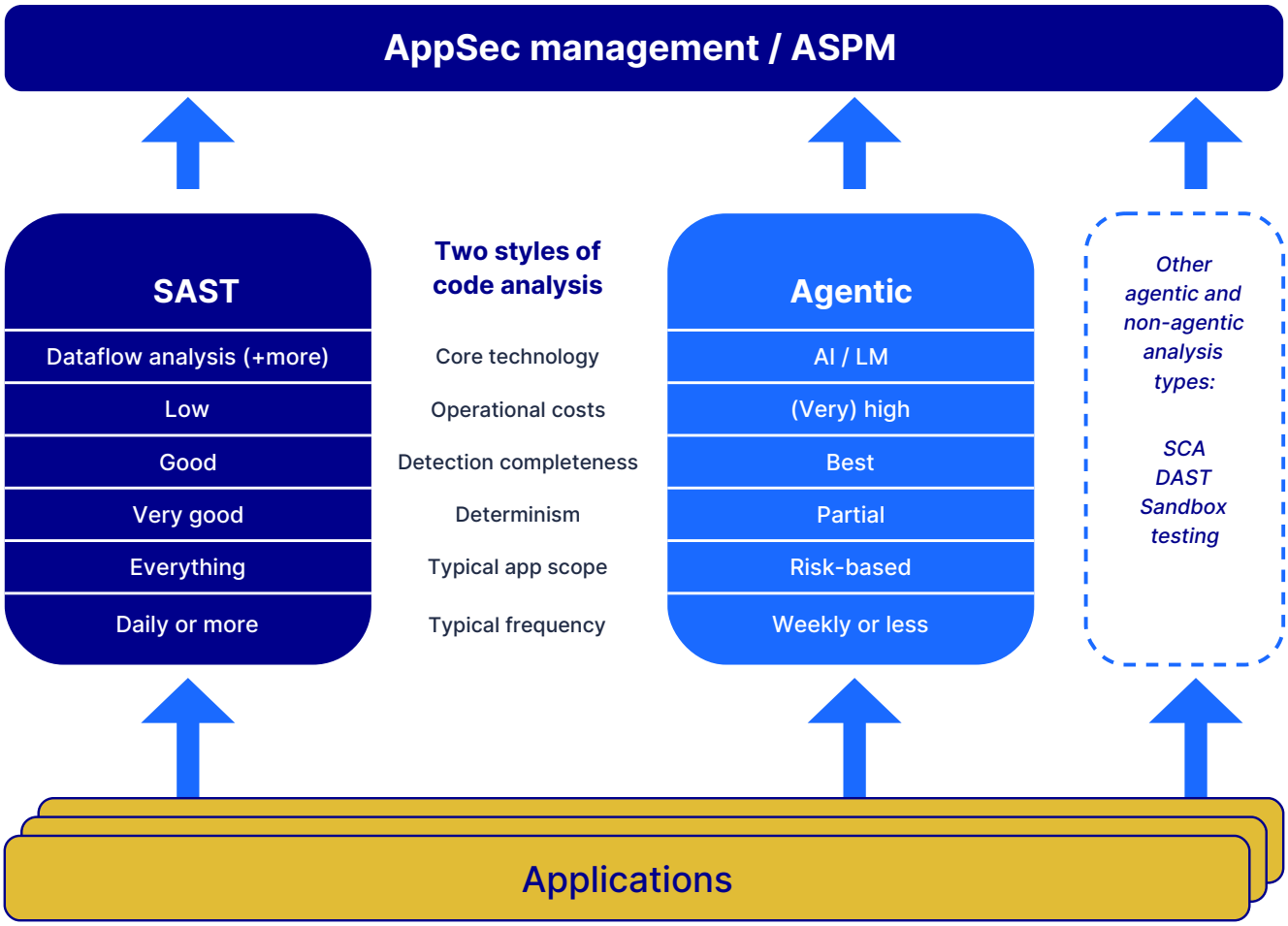
This is a real capability, not a roadmap slide. In a published study across 10 languages, FAA found 10 of 12 real-world CVEs, most rated critical (CVSS 9.0 or higher), all disclosed after the analysis model's training cutoff, so the results reflect reasoning about the code, not recall of a known bug. Fortify SAST gives you breadth, speed, and deterministic coverage; agentic analysis adds design-level reasoning about intent that no fixed set of rules can express. Together they cover ground neither reaches alone.[1]

Net-new findings, in one system of record

Fortify Agentic Analyzer runs as its own analyzer, and its findings come together with the rest of your Fortify results, not in a separate console or a new findings silo. It complements SAST rather than replacing it, deduplicating against your existing SAST findings to surface true blind spots. Findings are brought into Fortify™ Software Security Center and Fortify™ on Demand in the Fortify taxonomy, alongside Fortify SAST, Fortify™ DAST, and Fortify™ Software Composition Analysis.

How a scan works

A scan runs as a multi-stage agentic pipeline whose stages include planning (triaging and prioritizing the codebase), analysis (agents explore the code, form and adversarially test security hypotheses, and report structured findings), and validation (an independent pass re-verifies each finding, removing false positives and duplicates). A deterministic harness, not the agent, governs coverage, output stability, and budget adherence.



1 OpenText, *Applying agentic analysis to real AppSec problems*, July 2026

Resources

Fortify Agentic Analyzer finds real vulnerabilities
Watch the demo ›

Will AI replace application security testing?
Read the blog ›

Level up your agentic AppSec powers
Read the blog ›

Are AI labs coming for application security?
Read the blog ›

Fortify Agentic Analyzer
Learn more ›

Bring your own LLM, control your cost

FAA follows a bring-your-own-LLM model: Anthropic, OpenAI, AWS Bedrock, Google Vertex, Azure OpenAI, GitHub Copilot, or self-hosted, with per-stage model and effort selection. It is included under your existing Fortify SAST license; you pay your LLM provider for usage. Objectives are set through policy files (scope, stages, models, cost caps), so cost stays under your control.

Safe by design

The agent runs read-only and offline through strict tool allowlists (no write, edit, shell, or network access) and is instruction-isolated: agent instructions found in scanned code are treated as data, never as directives, and findings pass only through schema-validated reporting tools.

Agentic analysis only helps an enterprise if it lands where the team already works. Fortify Agentic Analyzer finds what traditional SAST cannot and files it into the same Fortify system of record you already run, under a license you already own. That is the line between a developer-side experiment and enterprise application security, and it is ground the pure-play AI scanners cannot claim. The future of application security is hybrid, and it is here now.

See how agentic analysis finds what your SAST scans miss, without adding another tool or another triage queue.

Contact us ›

Copyright © 2026 Open Text • 10.26 | 241-000253-002

opentext™