UNIFIED APPSEC RFP TEMPLATE FOR AI-DRIVEN DEVELOPMENT

snyk

Evaluating end-to-end security across the SDLC

For evaluating platform capabilities to secure every layer of modern application development.

Overview

From AI-generated code to cloud-native deployments, today's software environments are complex — and security needs to keep pace. Organizations are increasingly looking to consolidate fragmented tooling and standardize on a unified platform that can secure the entire SDLC (software development lifecycle), without slowing down innovation.

This template helps procurement and AppSec teams evaluate vendors across the full stack — code, open source, containers, infrastructure as code (IaC), and AI-generated components. It's designed to surface which vendors offer real breadth, performance, and governance capabilities, and which still rely on stitched-together solutions or manual effort.

Core capabilities to include in your RFP

01

COMPREHENSIVE TESTING COVERAGE

Support for multiple surfaces and artifact types.

WHAT TO ASK:

- Which types of security testing do you support natively (SAST, SCA, container, IaC)?
- Are testing engines built-in or stitched together through acquisitions?
- How do you ensure consistent findings and prioritization across surfaces?

02

DEVELOPER-FIRST INTEGRATION AND SCALE

Embed seamlessly into developer workflows across enterprise scale.

WHAT TO ASK:

- What IDEs, CLIs, SCMs, and CI/CD tools are supported?
- How does the platform perform across large-scale deployments (e.g., 1,000+ developers)?
- What mechanisms do you offer to reduce developer noise and prioritize critical issues?

03

AI-NATIVE AND GENAI-READY

Addressing modern threats introduced by AI-generated code and LLM-backed apps.

WHAT TO ASK:

- How does your platform handle security for AI-generated code?
- Does your solution offer an MCP server to integrate security testing directly into code generation tools and agentic workflows?
- Do you support an AI Bill of Materials (AI-BOM) to provide visibility into LLMs and agentic services?
- Are there protections against prompt injection, insecure agent orchestration, and AI model drift?

04

VERIFIED REMEDIATION AND AUTOMATION

Reducing time-to-remediate with safe, in-flow fixes.

WHAT TO ASK:

- Do you offer AI-generated code fixes that are validated for accuracy?
- How are these fixes delivered (e.g., in IDEs, PRs)?
- Can developers understand and trust these fixes through context-aware explanations?

05

POLICY, PRIORITIZATION, AND GOVERNANCE

Centralize control without creating bottlenecks.

WHAT TO ASK:

- How do you define and enforce application security policies across the SDLC?
- Can policies be written in natural language and adapt to changing risk contexts?
- Do you support governance across distributed teams and CI/CD pipelines?

06

BUILT-IN INTELLIGENCE AND DATA FOUNDATION

Security context backed by scale.

WHAT TO ASK:

- How is prioritization determined — does it factor in runtime, reachability, or business context?
- Do you leverage an aggregated data foundation or vulnerability database to inform decisions?
- How do you benchmark fix success and remediation time?

Side-by-side checklist: Key platform capabilities

CAPABILITY	VENDOR A	VENDOR B	VENDOR C
First-party engines for SAST, SCA, container, and IaC.	☐	☐	☐
Enterprise performance (1,000+ developers).	☐	☐	☐
IDE, CLI, PR, and CI/CD integration.	☐	☐	☐
GenAI and AI-native app security (AI BOM, prompt injection protection).	☐	☐	☐
Verified, context-aware remediation.	☐	☐	☐
AI-driven prioritization and noise reduction.	☐	☐	☐
Policy enforcement across teams and pipelines.	☐	☐	☐
Unified governance.	☐	☐	☐
MCP server to integrate security testing in AI tools and workflows.	☐	☐	☐

Tips for evaluators: Find substance over surface

The unified AppSec platform you choose will define your security posture for years to come. Look for:

- **True platform consolidation.** Don't settle for a bundle of tools with different interfaces and engines. Ask for a single policy engine and UI that works across the SDLC.
- **In-flow security.** Developers should be able to scan, prioritize, and fix without leaving their existing tools.
- **AI-native capabilities.** GenAI and LLM adoption is accelerating — your AppSec partner should already support them, not just have them on the roadmap.
- **Fix accuracy and automation.** Test the platform's ability to deliver verified, AI-powered fixes at scale.
- **Proven scale and performance.** Look for proof that the vendor can support large engineering teams with high-volume development velocity.

Built with insights from the Snyk AI Trust Platform

The Snyk platform combines first-party engines, verified AI fixes, and policy-driven governance to secure every layer of the software development lifecycle. Built for scale, speed, and AI-powered development. **Book a demo** to see unified AppSec in action.

snyk