

# SAST - Static Application Security Testing

 SOLUTION BRIEF



## SAST with Context

Static Application Security Testing (SAST) is used to identify vulnerabilities in custom application code and is often used early in the lifecycle before the application can be run.

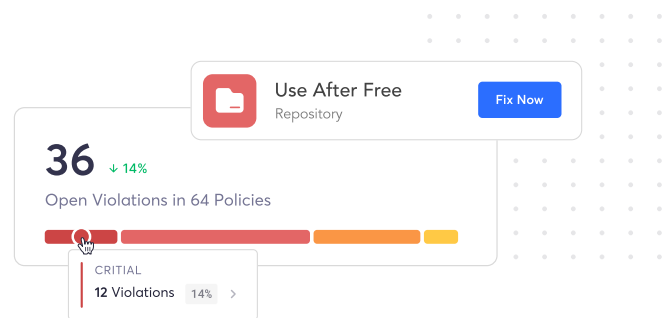
SAST promises early detection of vulnerabilities, but security teams struggle to translate this promise into improved security because existing SAST tools are too slow, inaccurate, and disruptive to modern development practices. These tools lack an understanding of each finding's context in the SDLC, forcing users to manually investigate every finding to effectively understand impact and priority. This wastes resources, disrupts developer workflows, reduces productivity, and results in haphazard management of security risks.

Cycode SAST avoids these problems with a highly accurate, lightning fast analysis engine that enables users to fix identified vulnerabilities seamlessly in established development workflows. More importantly, Cycode's deep understanding of the SDLC, from development to production, provides end-to-end context for AppSec findings so users don't need to research it manually. Cycode eliminates noise, produces deeper, more actionable insights, and enables teams to focus on the findings that represent the greatest risk. By supporting all popular, modern languages, Cycode minimizes the need for multiple SAST tools.

## Lightning Fast Scanning

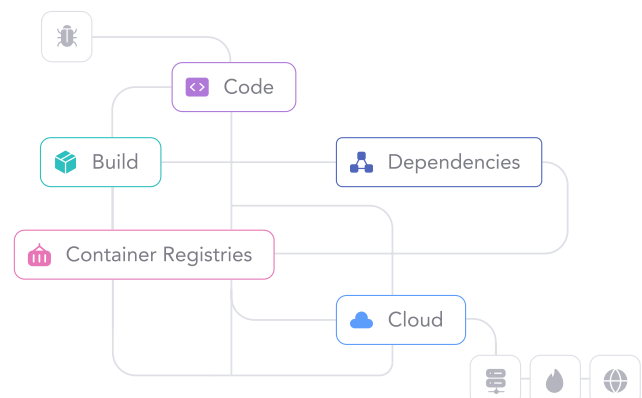
Modern SAST tools must fit into tight, automated DevOps processes and deliver results that are immediately actionable.

Cycode accelerates software delivery with a ridiculously fast SAST scanning engine that enables users to find and fix security vulnerabilities in custom application code without disrupting development.



## Accuracy from End-to-End Context

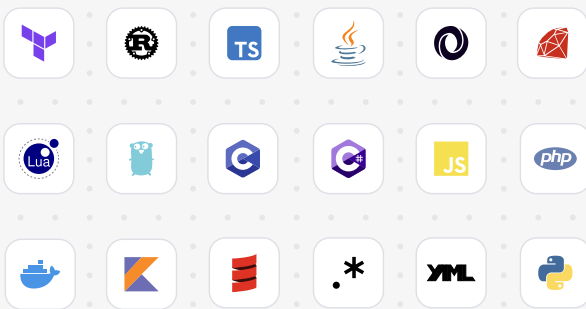
Stop wasting developers' time on findings that are not exploitable. The Cycode platform monitors development tools and activity to understand how code and artifacts flow through the SDLC, and connects these dots with SAST and other AppSec findings in the Knowledge Graph. With visibility from development to production, and end-to-end context for each AppSec finding, Cycode eliminates noise that your developers would otherwise waste time researching.



## Broad Language Coverage

Modern cloud native applications leverage loosely coupled services built on independent technology stacks aligned to the service domain or team expertise. This increases agility, but also complicates efforts to secure these applications.

Cycode supports both modern and legacy programming languages, including Java, C#, JavaScript, PHP, Python, and many more, so you can enforce security standards across all your apps from a single platform.



## Enable DevSecOps

Most DevSecOps initiatives fail due to long SAST scans, the need for manual investigation, and excessive time wasted on irrelevant findings. Without the ability to operationalize security controls in DevOps processes, teams cannot achieve DevSecOps.

Cycode avoids disruption with automated security tools that deliver prioritized, actionable findings to the right developers in their established workflows.



### Actionable Insights

Cycode delivers SAST results quickly, including end-to-end SDLC context that helps developers understand the

broader impact of findings and how they relate to other components and risks. There's no need for developers or security experts to manually investigate and prioritize findings before they can be acted upon. This translates into faster remediation and shorter backlogs.



## Developer Friendly Workflows

Designed from day one to provide an exceptional developer experience, Cycode integrates throughout the SDLC to meet

users in the workflows they already use. Automated integrations ensure that vulnerabilities are detected early and often, and that the right information is delivered to the right developer at the right time for efficient remediation. Regardless of whether they are committing code, reviewing pull requests, or working a ticket, developers never need to go out of their way to write secure code.



## Risk-Based Prioritization

Because Cycode has visibility from code to production, it understands your application, dependencies, CI/CD pipelines, runtime

environments, and the relationships between them. This information is used to provide intelligent prioritization based on advanced criteria such as deployment environment, whether findings are exposed and exploitable, and how much traffic they see. Developers can focus on and fix the most important issues faster.

## Complete Software Supply Chain Security

Eliminating vulnerabilities from custom application code is a necessary part of securing your software supply chain, but ultimately it's just one part of securing your application. Cycode SAST helps you secure your custom application code, and the Cycode platform secures the entire software supply chain from code to cloud so you can manage its vast and interconnected attack surface.

Cycode hardens your SDLC's security posture by implementing consistent governance, and reduces the risk of breaches with a series of scanning engines that look for issues like hardcoded secrets, code leaks, SCA, misconfigurations, SAST and more.

Cycode's Knowledge Graph tracks code integrity, user activity, and events across the SDLC to prioritize risk, find anomalies, and prevent code tampering.

 [Learn more at Cycode.com](https://cycode.com)