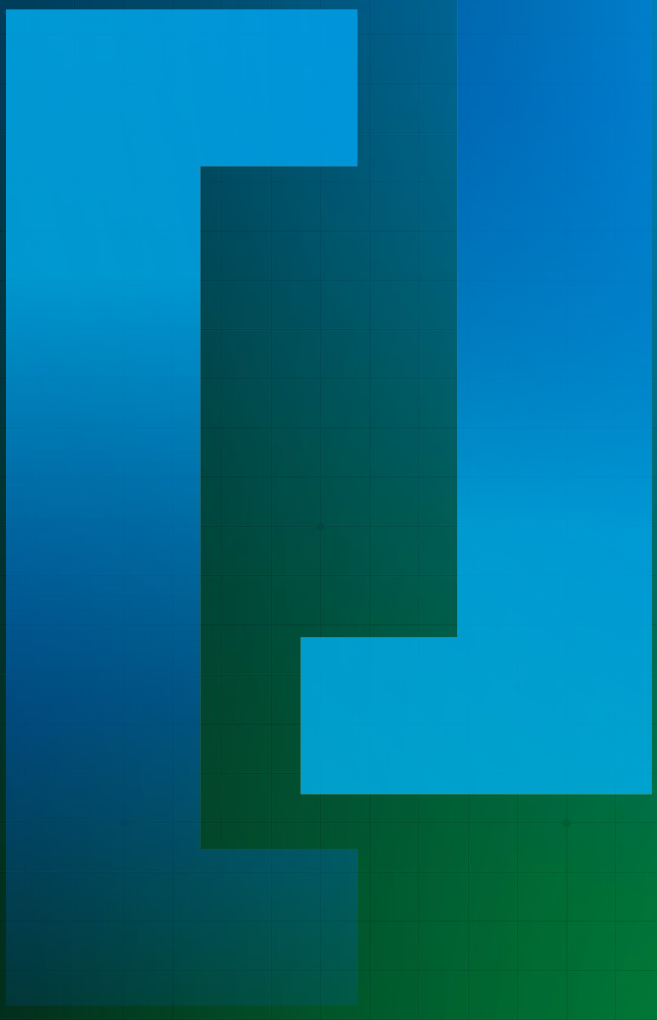
VERACODE

WHITEPAPER

The Next Generation of SAST Scanning

Veracode Adaptable Static Analysis: Technical Methodology, Operational Excellence, and Performance Analysis



Executive Summary

The Veracode SAST Engine represents nearly two decades of continuous security research and engineering excellence. Evolving beyond its foundational strength in binary analysis, the engine now powers Veracode's Adaptable SAST Scanning Service. This next-generation approach offers an effective scanning engine that supports source, binary, and hybrid (i.e., a combination of source and binary) analysis, eliminating the trade-off between speed and depth. Our next generation scanning service offers flexible configuration options to ensure it works the way you do.

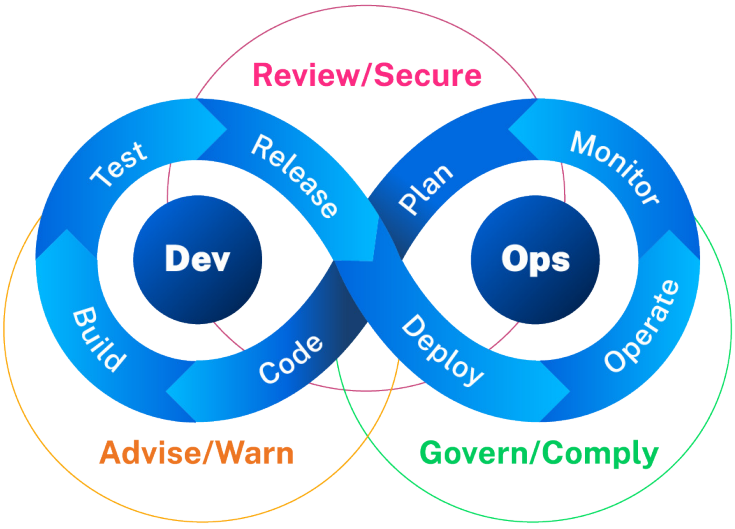
While maintaining rigorous accuracy using various proprietary techniques including its patented graph-based methodology, the engine is continuously evolving to meet developers exactly where they work. Our industry-leading static analysis scanner is being expanded to directly analyze Java source, in addition to Java binaries, providing the flexibility to choose source, binary, or both. By providing a source-code option, developers gain the best of both worlds: they can achieve a faster time-to-results by eliminating the early-stage build and packaging requirements, yet they still benefit from the same high-fidelity detection logic of our core SAST engine.

Pioneering SAST for over 20 years, we lead the industry by meeting developers where they work. As a Forrester Wave™ Leader in SAST with 9 perfect scores — more than any competitor — we seamlessly integrate into your process. The Veracode SAST scanner is architected to adapt to your specific use case and provide an effective scanning experience. We are continuously working to expand our Adaptable SAST Scanning Service with our upcoming Real-Time Interactive Scanning release. This capability serves as a real-time scanner centered on the Advise and Warn phase, delivering instant warnings that adapt to the developer's workflow. This interactive method is complemented by additional phases to ensure a secure, auditable development lifecycle:

The #1 Leader in Precision

Awarded 9 perfect scores in the Forrester Wave™.

- **Advise & Warn:** Delivering real-time warnings and inline remediation guidance directly in the IDE as code is edited. This "client-side" intelligence provides instant feedback, allowing developers to catch and fix security flaws before they are even committed to the repository.
- **Review & Secure:** Performs a comprehensive scan upon commit or merge request to ensure code is secure before it is merged providing a verification layer of findings, ensuring an uninterrupted workflow.
- **Govern & Comply:** Culminating in definitive policy scans that enforce industry standards and organizational mandates, providing the auditable results required for comprehensive security compliance.



Our unified architecture offers a range of options whether a developer needs a lightweight scan for rapid iteration all the way to a deep scan for final verification, or something in-between, the results are always derived from the same deterministic, high-fidelity detection logic.

This whitepaper details the technical architecture, testing rigor, and operational safeguards that power the SAST engine. It specifically addresses critical questions regarding the engine's efficacy, cross-language stability, severity classification, and data security.

Veracode SAST Engine Architecture

Deterministic Graph-Based Analysis

The SAST engine does not "guess" at vulnerabilities using probabilistic machine learning. Instead, it builds a comprehensive internal representation or model — specifically, a Semantic Graph — that represents the application's entire execution logic and data paths.

- **Control Flow Analysis:** The engine maps every possible execution path, analyzing conditional branches, loops, and exception handling to determine code reachability.
- **Data Flow Analysis:** It tracks data movements from "sources" (user inputs, API calls) to "sinks" (databases, file systems), identifying exactly where untrusted data interacts with critical functions.

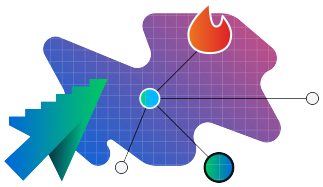
> **Note on "Models":** In the context of the Veracode SAST Engine, a "model" is a precise, deterministic representation of program behavior, not a stochastic machine learning model. Given identical input, the engine produces identical output every time (100% reproducibility), requiring no statistical confidence intervals.

Language-Agnostic Internal Representation (IR)

To ensure stability across different technologies, the SAST engine converts all supported languages (Java, .NET, Python, JavaScript, etc.) into a Common Internal Representation (IR) before analysis begins. This ensures that the same high-fidelity detection logic is applied uniformly to every scan, regardless of the programming languages and frameworks that are used.

Testing Methodology & Quality Assurance

The SAST engine's accuracy is maintained through a rigorous, multi-layered quality assurance framework that governs every monthly release.



The Testing Ecosystem

The engine is validated against a massive internal suite to ensure no regressions occur:

- **Internal Test Suite (100,000+ Applications):** The engine is tested against a comprehensive suite designed to represent the wide range of operating systems, languages, compilers, and versions supported.
- **Annotated Test Cases:** Thousands of specific test cases, written or reviewed by the Applied Security Research Team, are annotated with expected findings (CWEs), line numbers, and severity ratings to verify detection accuracy.
- **TrackQ Program (Real-World Validation):** Veracode partners with select customers who consent to permission to test engine updates against real-world production applications. This ensures that the engine's performance is validated against actual enterprise coding patterns, not just synthetic tests.

Release Validation & Confidence

Every change to the SAST engine undergoes "Pre-Merge" testing, followed by an exhaustive "Release Validation" cycle.

- **Methodology:** The primary validation method is *Baseline Comparison*. The new engine release is run against the current production baseline. Any difference in findings (new flaws or removed flaws) is manually analyzed to confirm it is an intentional improvement (e.g., a corrected rule or new coverage) rather than a regression.
- **Engine Performance & Continuous Improvement:** Our engine undergoes a rigorous monthly release cycle to ensure peak efficacy. Recent updates include validated improvements in third-party code detection for C/C++ and Java, enhanced COBOL parsing, and a reduction in false positives for credential management (CWE-259/798).

Stability Analysis: Cross-Language Consistency

A frequent concern is whether the SAST engine is equally effective across different languages (e.g., Does a Python app receive the same scrutiny as a Java app?).

The SAST Engine provides consistent effectiveness across languages. Because of the Common Internal Representation (IR), the detection algorithms operate on the structure of the code, not the specific syntax of the language.

- **Uniform Detection Logic:** A SQL Injection pattern (CWE-89) is identified by the same fundamental data flow violation (untrusted data reaching a SQL sink) whether it occurs in a Python Django application or a Java Spring Boot application.
- **Language-Specific Nuance:** While the logic is consistent, the engine is tuned to understand language-specific idioms (e.g., Java's JDBC calls vs. Python's ORM methods) and safety features.
- **Outcome:** If equivalent vulnerabilities exist in a Python app and a Java app, the SAST engine will detect both with consistent severity ratings. The efficacy does not degrade simply because the language changes.

Severity Ratings & Detection Rules

CWE Alignment

The SAST engine aligns its findings strictly with the Common Weakness Enumeration (CWE) standard.

- **Verification:** Severity ratings are verified during the QA process using Annotated Test Cases. If a test case is marked as a "High" severity SQL Injection, the engine is tested to ensure it produces a finding that matches that specific severity and CWE ID.
- **Updates:** The scanner is updated within 90 days of any new CWE release to ensure compliance.

Proprietary vs. Standard Rules

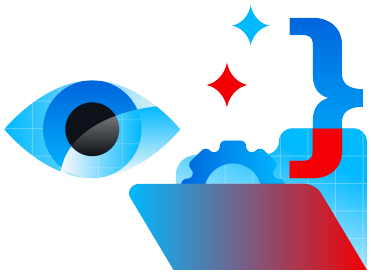
While the SAST engine maps findings to standard CWEs, the methods used to identify them include proprietary logic:

- **Proprietary Rules:** The engine utilizes proprietary algorithms to identify complex vulnerability patterns and framework-specific issues (e.g., React, Angular, Spring Boot) that may not be fully covered by public standards.
- **Advanced Cleansing Recognition:** To minimize false positives, the engine utilizes a proprietary database of recognized cleansing functions or sanitizers. These are specialized code routines — such as HTML encoders or SQL parameterizers — designed to neutralize malicious input. The engine intelligently recognizes when untrusted data has passed through a valid sanitizer, allowing it to accurately distinguish between a genuine vulnerability and code that has already been secured. This level of insight prevents the "noise" common in other tools that fail to account for existing security controls.

Benchmarking & Performance

The Veracode SAST Engine prioritizes internal consistency and rigorous regression testing over external "grading" benchmarks, which can be subjective.

- **Benchmarking Methodology:** The engine's performance is benchmarked against its own previous versions to measure:
  - **Runtime Performance:** Scan completion times and efficiency.
  - **Memory Usage:** Stability during large application scans.
  - **Finding Accuracy:** Precision of results compared to the "Gold Standard" annotated baseline.
- **External Validation:** The SAST engine consistently achieves leadership positions in independent analyst reports, such as the VDC Research 2025 Survey where Veracode Static Analysis is named a leader for its effectiveness in finding and fixing code flaws. We are also name a leader in the Forrester SAST Wave™ based on a rigorous evaluation of current offerings, strategy, and customer feedback.



Recognized as a SAST Leader

by Independent Analysts

Data Security & Multi-Tenancy

Veracode operates the SAST engine within a secure, multi-tenant SaaS platform designed for enterprise data isolation.

- **Tenant Isolation:** While the engine operates in a multi-tenant environment, data is logically separated. Customer code submitted to the SAST engine is never co-mingled; there is no cross-pollination of customer data or results between clients.
- **No Cross-Customer Learning:** The SAST engine does not train detection models on customer code. Its deterministic nature means it does not need to "learn" from one customer's code to find bugs in another's.
- **Encryption:** All data analyzed by the engine is encrypted in transit and at rest.
- **Residency:** Regional data residency options (e.g., European Region) are available, ensuring the engine processes data within specific geographic boundaries if required.
- **Data Retention & Disposal:** Minimize risk through retention policies. Source and binary artifacts are automatically purged after 45–90 days, while scan metadata is retained for compliance auditing.

VERACODE

The Strategic Choice

The Veracode SAST engine is a mature, deterministic solution that prioritizes accuracy, reproducibility, and flexibility. By offering an adaptable service that encompasses source, binary, and hybrid scanning, the engine matches the speed of modern development while delivering the deep security assurance enterprises require.

Request a demo today!
or learn more at www.veracode.com