Checkmarx

SOLUTION BRIEF

Checkmarx Fusion

The Most Comprehensive, Highest-Fidelity Scanner. Model-Agnostic by Design

Security teams are too often forced into the same difficult tradeoff. When a scanner is tuned to find everything, teams are overwhelmed by false positives. When its sensitivity is loosened to reduce that noise, genuine vulnerabilities can slip through, sometimes with serious consequences that only become clear later.

Checkmarx Fusion is purpose-built to eliminate that compromise. Depending on the language, either a rules-based engine or an AI-based engine runs. The Findings Analysis Engine reconciles that output into one clean result set. A multi-model AI engine then runs several curated frontier models on top, surfacing vulnerabilities no rule, CVE, or signature could catch. The result is a scanner that finds more of the exploitable vulnerabilities that matter and generates far fewer false alarms without the unpredictable token costs of running frontier models directly. Checkmarx Fusion is model-agnostic by design: Checkmarx curates and validates the models, so customers choose the models that fit their needs the best and get consistent, research-backed results at a predictable cost, without being locked to any single provider. In addition, the more scans run, the better performance: incremental scanning means faster scans over time, while optimizing costs.

The Challenge

That tradeoff comes from relying on one detection method at a time. Rules-based engines are precise and explainable, but they only catch what they've been explicitly written to look for. Anything requiring logic to identify or exploit slips past them. A vulnerability only visible when you trace how code will actually behave at runtime, and not just what it contains, is invisible to a rules-based scanner. AI-generated code makes this worse. AI-based engines catch that novel logic, but they also introduce more false positives and less consistency than a team can act on.

Most scanners run on one engine type, not both – so teams get that approach's strengths and blind spots as a package deal.

This single-engine tradeoff does not stop at one tool. It repeats across the application security stack. Organizations often use separate scanners for SAST, secrets, Infrastructure-as-Code (IaC), and APIs, and each one brings its own version of the same problem.

One tool may miss novel logic in AI-generated code. Another may bury secrets findings in noise. Another may flag infrastructure misconfigurations without making it clear which ones actually matter. The result is a fragmented view of risk, with security teams forced to stitch findings together manually, compare outputs across tools, and decide which alerts deserve attention. Those gaps are where real risk gets missed.

The cost of missing something is rising quickly. Defenders used to have more time to respond. WannaCry was weaponized in 59 days after its patch shipped, and analysis of 25 historical vulnerabilities found that 16 of them took a month or more to exploit. That window has now collapsed. Anthropic's own research found that Claude Mythos, its most capable model, produced a working Firefox exploit within an hour of Mozilla issuing the patch – while the patched browser itself took 18 days to reach users. The time between disclosure and weaponization that security teams used to plan around no longer exists.

Security teams need accurate, trusted findings sooner – with less manual effort and fewer opportunities for real risk to slip through the cracks.

The Solution

Security teams do not need a scanner that simply leans further into one detection method. They need an approach that closes the gap between methods without forcing a tradeoff between them.

That is what Checkmarx Fusion is designed to do. It combines rules-based deterministic detection and AI-based detection in a single architecture.

The Findings Analysis Engine (FAE) analyzes those results to determine what is real risk and what is just noise. It then consolidates both engines' outputs into a single list, so teams do not receive separate, conflicting sets of alerts.

The impact is clear. This pipeline isn't a compromise between two imperfect approaches. Instead, it combines the strengths of both deterministic and AI-based reasoning. Because Checkmarx Fusion runs a bounded, curated set of Checkmarx-validated models rather than open-ended frontier API calls, security teams get consistent results at a predictable cost, not a token bill that compounds across every commit and pull request.

This can be measured by the F1 score. Precision asks: of all the findings reported, how many are real? Recall asks: of all the real vulnerabilities in the code, how many were actually found? F1 captures both at once, so a high score means you're finding what matters without burying the team in noise.

After implementing Checkmarx Fusion, Checkmarx SAST's F1 score rose from 0.64 to 0.74. That is not a marginal gain. Moving from 0.64 to 0.74 means catching a meaningfully larger share of exploitable vulnerabilities that rules-based scanning misses, including the ones most likely to become a breach. This improvement rate is almost four times higher than the industry average and reduces false positives by 60–70% compared with standalone AI scanning.

Detection accuracy (F1 score), each tier built on the one before it.

0.20	0.64	0.74*
Query-based SAST	Next-Gen SAST	Checkmarx Fusion

**Analysis of Checkmarx data, Checkmarx Zero, 2026.*

Checkmarx's New Hybrid Architecture

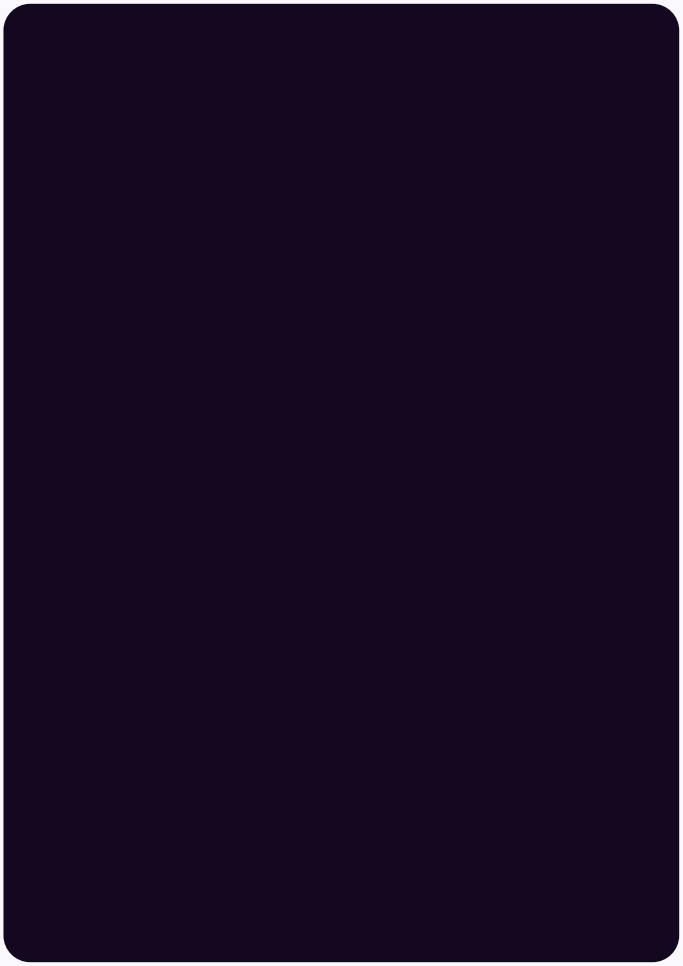
- **Deterministic Engine:** Rules-based analysis of known vulnerability classes. Precise, explainable, and consistent from scan to scan.
- **AI-Based Engine:** Extends coverage to languages and patterns that rules alone cannot reach, including novel logic and AI-generated code.
- **Findings Analysis Engine (FAE):** Reconciles the output of both engines into one clean result set. False positives removed, true positives kept.
- **Multi-Model AI Engine:** The multi-engine AI layer on top. Several curated frontier models reason about the code, each approaching it differently. Together they surface vulnerabilities with no known rule, no prior CVE, and no signature to match against. The models are Checkmarx-curated and research-validated. Models are flexible, so users can control costs by choosing either high- or low-cost models, according to their needs.

Connected to the Rest of the Risk Picture

That same architecture is designed to extend across SAST, Secrets Detection, IaC, and API security – so findings don't end up scattered across four different tools. Instead, they flow into Risk Orchestration, part of Checkmarx One ASPM, where a SAST finding can be assessed alongside an IaC finding, a secrets finding, or an API security finding, all scored and prioritized through the same risk-based lens.

That consistency matters across every scan type. When teams can trust the fidelity of one engine, they can trust the next one the same way, without having to relearn what counts as noise for each new part of the pipeline.

That trust is what makes the next step possible. Once results are consistently high-fidelity, they're reliable enough to hand off to agentic tools, so Triage Assist and Remediation Assist can turn findings into prioritized decisions and merge-ready fixes with confidence. That means less noise to review, less time spent deciding what is real, and more time focused on producing merge-ready fixes – all with a human still reviewing and approving before anything ships.



Key Features

1 One architecture

The same deterministic-plus-AI approach runs underneath every scan type, instead of a different bolted-on tool for code, secrets, infrastructure, and APIs.

2 The numbers hold up

In SAST, query-based scanning scores 0.20. Next-Gen SAST reaches 0.64. Checkmarx Fusion gets to 0.74, with 60 to 70 percent fewer false positives than standalone AI scanning.

3 One risk picture

Every scan's findings land in Checkmarx One ASPM next to each other, scored against the same risk model.

4 The same answer every time

Scan it twice, get the same result, on any scan type.

5 Stands up to scrutiny

Every finding traces back to a rule or a verifiable signal, the same standard in SAST as in every other scan type.

6 Performance

Incremental scanning means faster and faster scans over time.

7 Cost optimization

Prompt, context, and harness tuning and incremental scans.

8 Model optionality

High- or low-cost models, chosen per need.

9 Built to keep adding

Secrets Detection, IaC, and API security follow the same architecture, so each new scan type strengthens the same risk picture instead of starting a new one.

About Checkmarx

Checkmarx is the leader in agentic application security, delivering enterprise-grade protection while lowering engineering costs and accelerating development velocity. The Checkmarx One platform scans trillions of lines of code each year for companies, cutting vulnerability density by more than half. Its autonomous security agents detect and counter AI-driven threats across the SDLC, providing prevention-first protection for legacy, modern, and AI-generated code at enterprise scale. Follow Checkmarx on LinkedIn, YouTube, and X.

Discover Checkmarx Fusion.

Request a Demo

Checkmarx

AppSec for the Age of AI-Generated Risk

One Platform. Every Surface. One Risk View.

Checkmarx One brings security into every stage of development, so teams ship faster without tradeoffs. AI agents find and fix issues in real time, hybrid scanning covers every attack surface, and unified risk intelligence delivers contextual insights.

CHECKMARX ONE

CAPTURE ALL RISKS
Fusion Scanning Architecture

- Code Security
- Supply Chain Security
- AI Supply Chain Security
- Runtime Security

ELIMINATE & PREVENT RISKS
Assist Family of Agents

- Developer Assist
- Triage Assist
- Remediation Assist
- Checkmarx MCP

GOVERN & UNIFY RISKS
Orchestration/Posture Management

- Risk Orchestration
- Analytics & Reporting
- Policy Management

www.checkmarx.com