The Vulnerabilities Were Always There Now What?

How To Secure Yesterday's Debt and Getting Ahead Of AI-Generated Code Risk

Jonathan Rende
Chief Product Office, Checkmarx

Eran Kinsbruner
VP Product Marketing, Checkmarx

Agentic AppSec Unleashed '26
by Checkmarx

Agenda

The Volume and Backlog Changed

The Economics of Exploitation

The Impact on AppSec and Engineering

Can AI Help?

The Right Approach

AI-GENERATED CODE AS A RISK CLASS

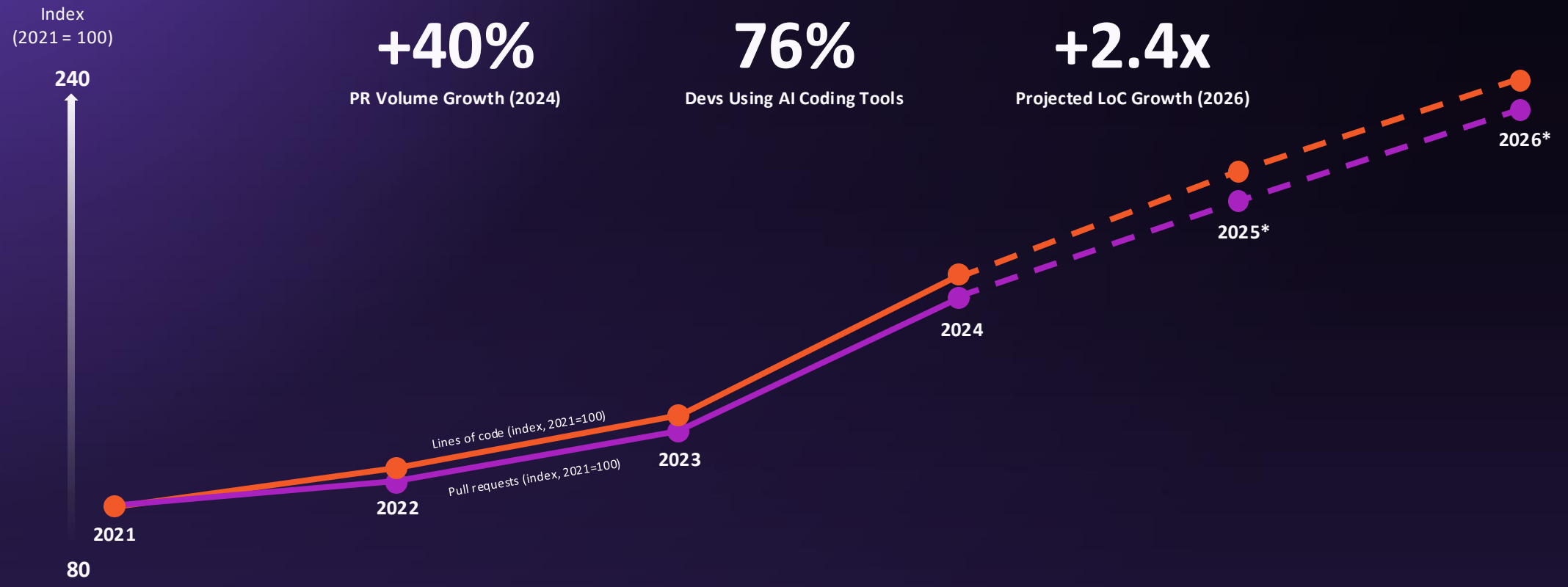
Volume of AI Generated Code And Risk Going Up

Even the Best Frontier Model Ships ~46% of Solutions Either Incorrect or Insecure

+40%
PR Volume Growth (2024)

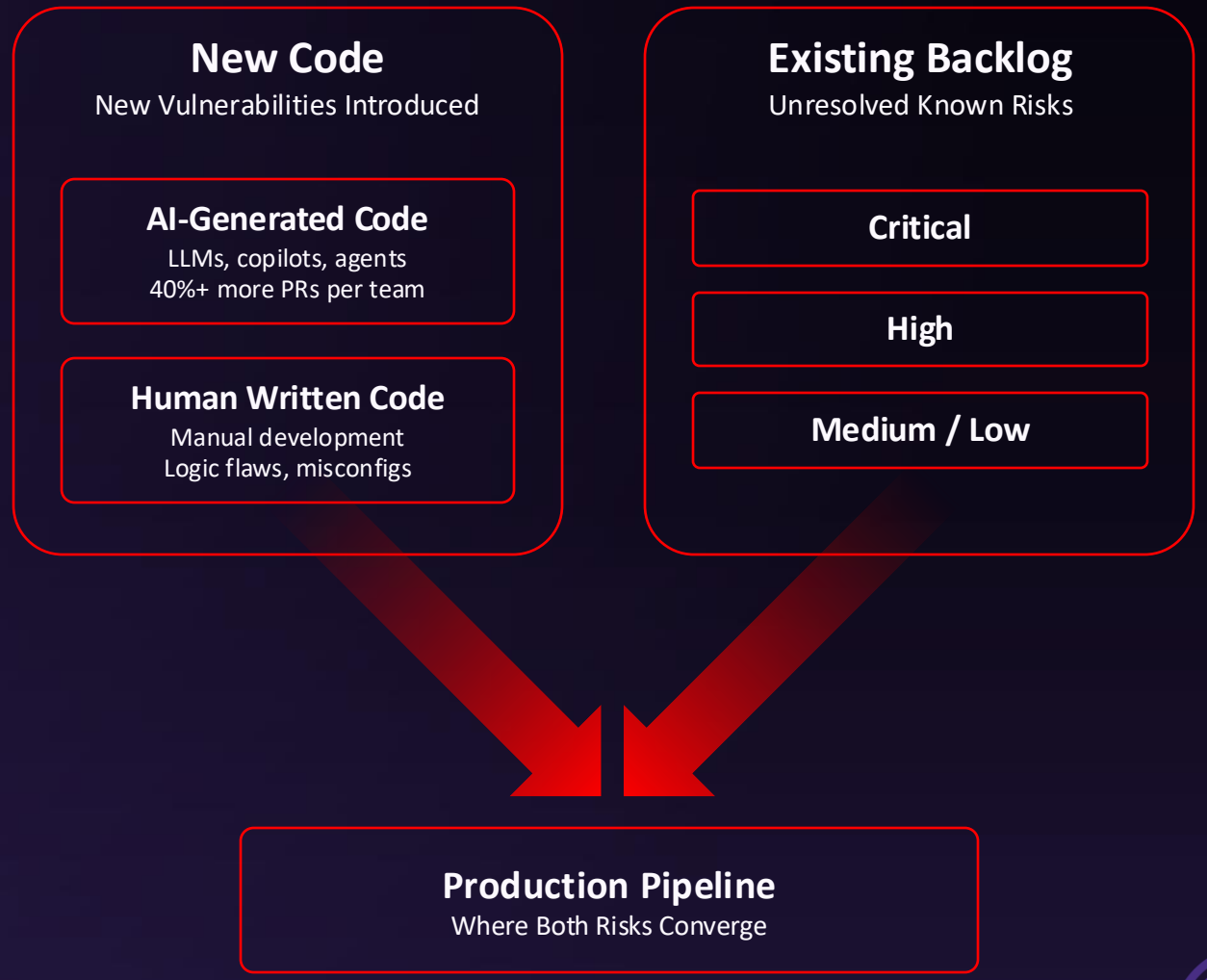
76%
Devs Using AI Coding Tools

+2.4x
Projected LoC Growth (2026)



Sources: GitHub Octoverse 2024, GitLab DevSecOps Survey 2024, Stack Overflow Developer Survey 2024. 2025–2026 values are projections based on observed CAGR. Dashed lines indicate projected data.

Where Vulnerabilities Exist and Where They're Coming From



Agentic AppSec Unleashed '26
by Checkmarx

PART 1

The Economics of Exploitation

Permanently. In 2026.

AI-GENERATED CODE AS A RISK CLASS

The Hidden Cost of AI-Assisted Development

1.7x

More defects per unit of code in AI-assisted dev.

5x

Net exploitable flaws when 2–3× code volume meets 1.7× defect rate.

81%

Of orgs knowingly ship code with known vulns. Triage overload is cited cause.

Sources: BaxBench · CodeRabbit State of AI vs. Human Code Generation · Checkmarx

Weaponization: From 840 Days to 1.6 Days

AI-Generated CVE Exploits Cost ~$1 and 10–15 Minutes of Compute

525x

Compression in time-to-exploit since 2018

Projected 2028:

Under 1 Minute

Source: Cryptika, 2026 · Median time-to-exploitation of disclosed CVEs.

Checkmarx

THE UNIT ECONOMICS OF ATTACK

Discovery is No Longer the Constraint

~$1

Cost per working CVE exploit
10–15 min of compute,
fully automated.

72.4%

Frontier AI exploit success rate
On real-world targets. Near-zero
for prior generations..

1.6 Days

Median time-to-exploitation, 2026
Was 840 days in 2018. Patch
Tuesday is no longer a defense.

Agentic AppSec Unleashed '26
by Checkmarx

PART TWO

Impact:
New Code and Risk is Accelerating While the Backlog is Filling Faster Than We Can Drain It

THE REMEDIATION BIFURCATION

Findings Up 76%. Fixes Down 46%.

THE DASHBOARD is Misleading

Discovery Scaled with AI, but Remediation Didn't

−80%

Mean time to remediate (MTTR)

−73%

Critical MTTR

WHAT'S ACTUALLY HAPPENING

The Exposure Debt

+76%

Vulnerability submissions, last 12mo

−46%

Monthly fixes shipped, same period

25×

Growth in unresolved critical vulns

Source: HackerOne / The Weather Report — "Finding Fast, Fixing Slow," May 2026.

Agentic AppSec Unleashed '26
by Checkmarx

PART THREE

So, Can AI Help?

THE STRUCTURAL ARGUMENT

Separation of Church and State is Critical For Cybersecurity

The Security Control Plane Must be Architecturally Independent of the AI Systems it Governs

Access Control

A system that manages its own permissions cannot be trusted. A compromised system grants itself elevated access.

Cryptography

A key management system that stores its own master keys is insecure by design.

Audit and Finance

External auditors are required by law — not because auditors are dishonest, but because structural integrity demands independence.

DESIGN PRINCIPLES

Three Properties of an Independent Control Plane

If a Security Architecture Lacks Any of These, It Inherits the Failure Modes of the System It's Supposed to Govern

1

Architecturally Separated

Operates from outside the trust boundary of the AI systems it governs. Compromise of the model under analysis does not compromise the analyzer.

2

Deterministic Floor

Includes a rule-based layer whose behavior is defined by explicit, auditable logic — not probabilistic inference. The floor is what cannot be hallucinated.

3

Model-Agnostic

Capable of governing AI systems regardless of which foundation model underlies them. Single-provider concentration is an architectural risk.

Agentic AppSec Unleashed '26
by Checkmarx

PART FOUR

The Building Blocks for A Governance Framework for the Velocity We Have

What's Needed:
An Agentic Orchestration Layer and Security for AI

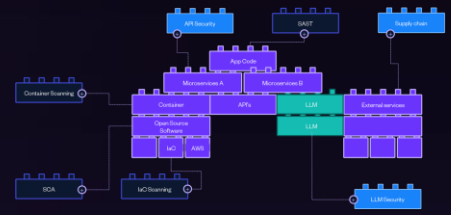
AI for Security (for Devs)

Design & Architecture → Coding In IDE → Code Commit Local Git repo → Pull Request

AI for Security (for AppSec)

Code Review & Merge → Build Process → Deploy Via orchestration → Go-live Monitor & Maintain

Security for AI



Autonomous and human assist agents operating within the AI stack across the lifecycle at key control points

Governance and scanning for all the new AI threat vectors

What's Needed:
A Hybrid Approach for Complete Coverage

Deterministic Ground Truth + AI-Augmented Reasoning + Security Context. None Alone is Sufficient.

Delivers

The highest possible fidelity
Consistent scanning results
High velocity remediation
Complete visibility

EMPIRICAL VALIDATION

What's Needed: A Hybrid Approach for the Highest Fidelity

Checkmarx SAST vs. Claude Opus 4.7

747 Findings · Python, Go, C# · Production Codebases (Istio, Indico, Jellyfin) + ThreatByte

+11%

Higher true positive rate (Checkmarx SAST)

327

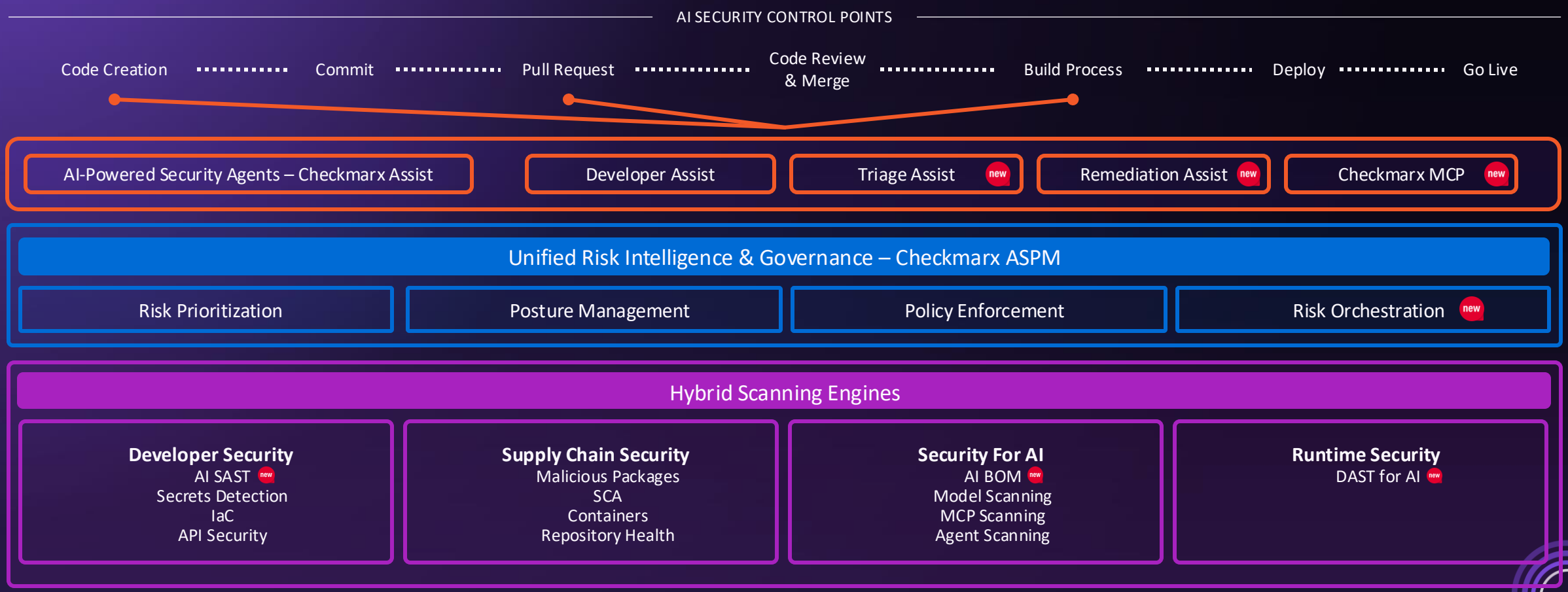
True positives Opus 4.7 missed entirely

0.49 F1 Score

Opus 4.7 and average SAST tools are at ~0.20

Opus 4.7 performed adequately only on ThreatByte — a vulnerable-by-design test project. Accuracy degraded across every production codebase. This is not a capability gap. The next model generation does not close it.

Checkmarx One: Unifying Application Security for the AI Era



THE CHECKMARX ADVANTAGE

Five Pillars. One Platform. Zero Compromise.

The Only Platform That Combines Deterministic Precision, AI-Powered Discovery, and Enterprise-Scale Governance — Unified in Checkmarx One.

Completeness of Findings

Hybrid deterministic + LLM coverage across every attack surface — SAST, SCA, DAST, IaC, secrets, containers. N-day precision. Zero-day discovery. The superset of findings.

Highest Fidelity

Correlated findings with the highest true-positive rate and lowest false positives. ASPM prioritization by real exploitability and business impact — not raw severity.

Guided and Autonomous Remediation

Best-in-class agents across IDE, CLI, and PR close the weaponization window. Human-in-the-loop governance ensures speed without introducing new risk.

Security Context as First-Class

20 years of AppSec expertise. 1 trillion lines of analyzed code. The largest malicious OSS package database. Training foundations no frontier LLM can replicate.

Governance, Compliance and Policy

Built for enterprise scale — robust policy management, customizable guardrails, and enterprise-grade IAM. Designed to scale AppSec programs, not just individual developers.

Scan-to-fix velocity is the goal — shifting from detection to prevention and remediation of new code issues is what will reduce your risky backlogs.

Discovery is no longer the binding constraint.

Remediation throughput is.

Agentic AppSec Unleashed '26

by Checkmarx