



**AppSec Platform for the
Software Development Revolution**

The AI Wave Is Drowning Security

84%

**of developers use AI
to build software**

According to 34k respondents on a
StackOverflow survey

62%

**of AI generated code
has vulnerabilities**

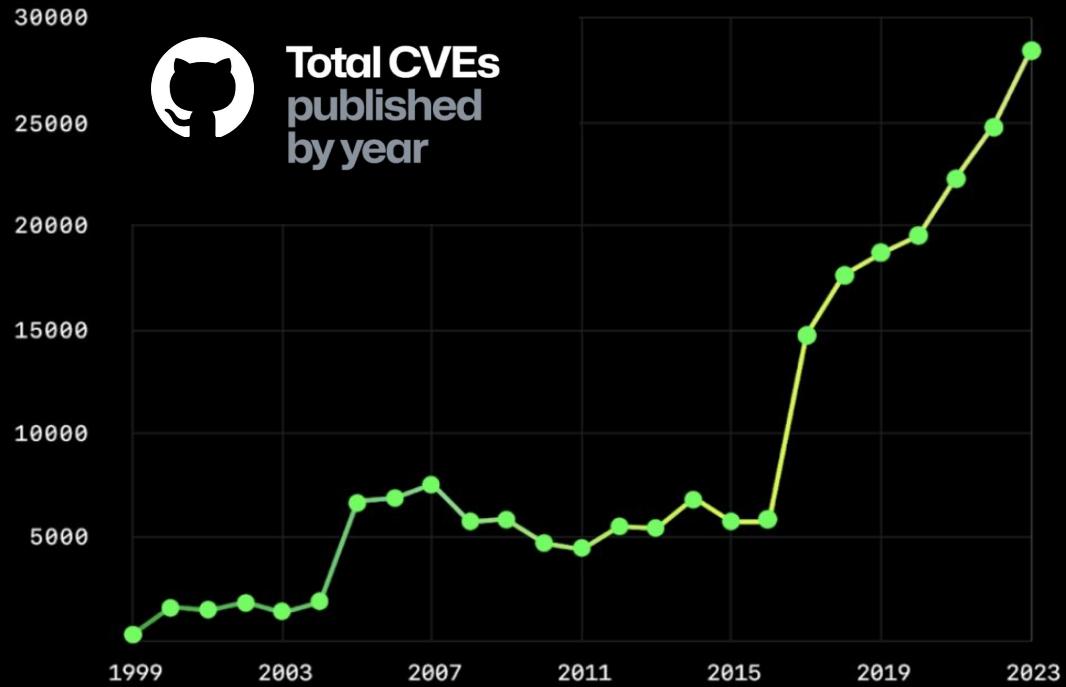
According to the research paper
“BAXBENCH: Can LLMs Generate
Correct and Secure Backends?”

42%

**of breaches exploit
web applications**

According to Verizon DBIR and
Mandiant's M-Trends

The Familiar Problems are Made Worse



Source: IDC, The Hidden Cost of DevSecOps, 2024

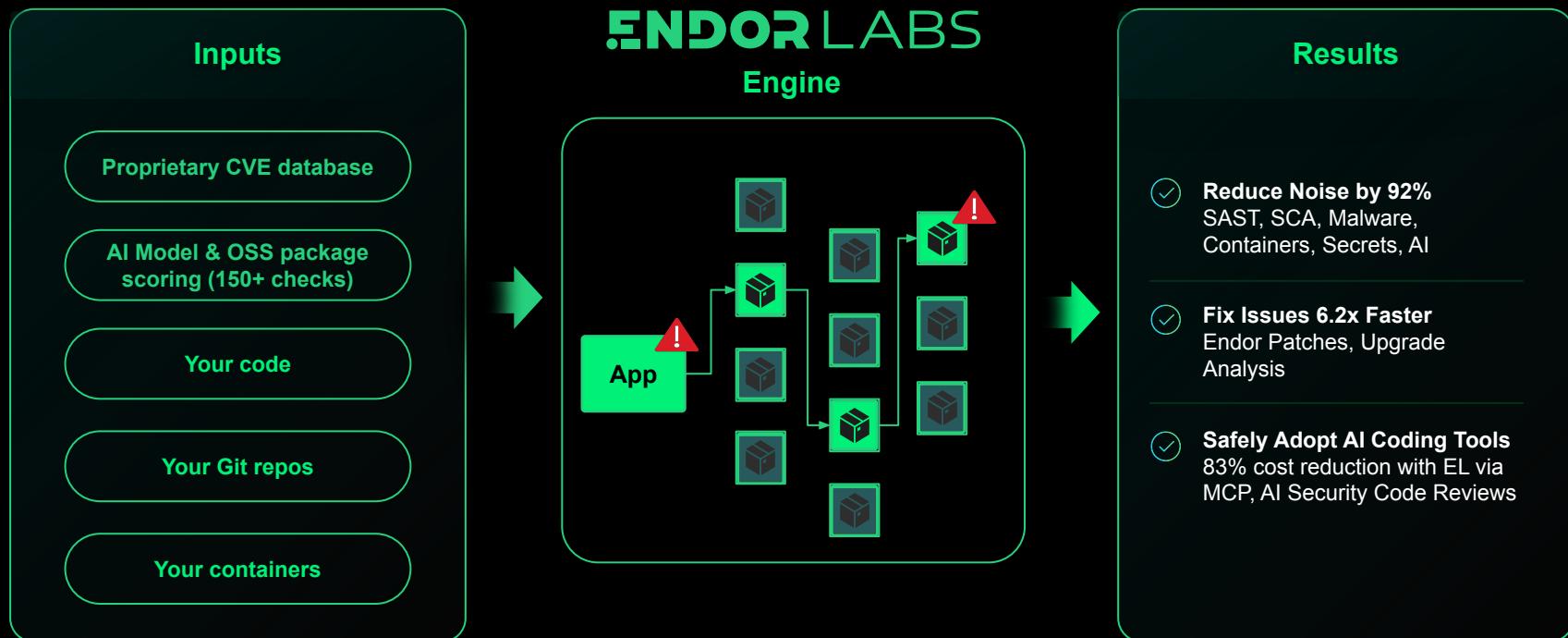
Too Much
Noise

Too Many
Tools

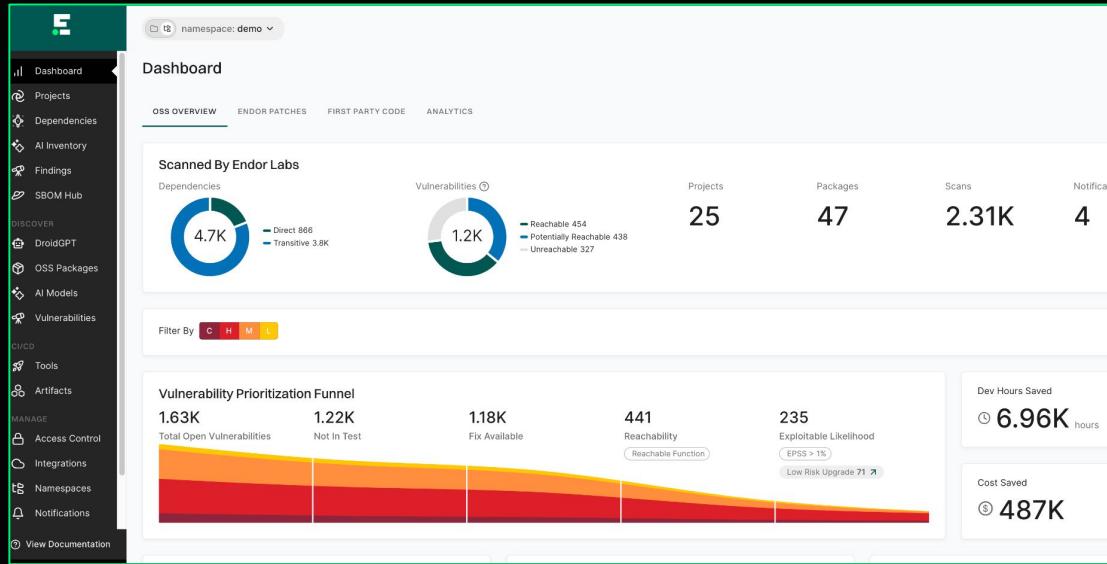
Too Long to
Remediate

Analyze **every** line of code, **every** dependency, on **every** layer

Whether it was written by humans or AI



Find What Matters With High Precision Code Scanning



The dashboard provides a high-level overview of the scanned code base. It shows the following metrics:

- Scanned By Endor Labs:**
 - Dependencies:** 4.7K (Direct 866, Transitive 3.8K)
 - Vulnerabilities:** 1.2K (Reachable 454, Potentially Reachable 438, Unreachable 327)
- Projects:** 25
- Packages:** 47
- Scans:** 2.31K
- Notifications:** 4

Vulnerability Prioritization Funnel:

Category	Value
Total Open Vulnerabilities	1.63K
Not In Test	1.22K
Fix Available	1.18K
Reachability	441 (Reachable Function)
Exploitability Likelihood	235 (EPSS > 1%)
Low Risk Upgrade	71

Cost Savings:

- Dev Hours Saved: 6.96K hours
- Cost Saved: \$ 487K



Best-In-Class Security Scanners

Reachability-based SCA

SAST

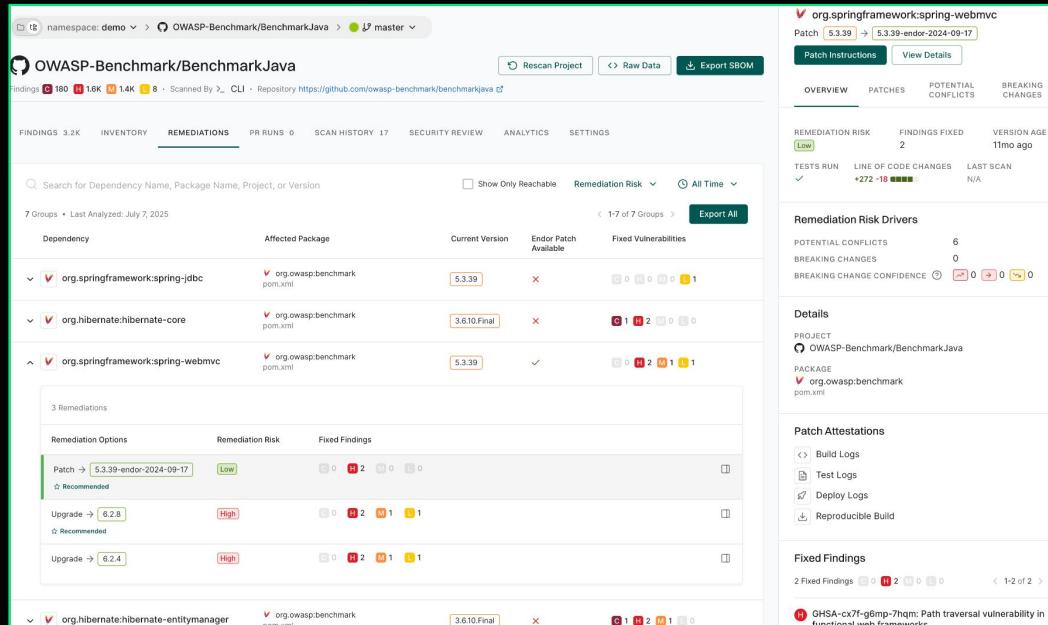
Container Scanning

Secret Scanning

AI Model Discovery

Malicious Dependencies

Fix What Matters With Security Patches and Upgrade Impact Analysis



The screenshot shows the OWASP Benchmark tool interface. At the top, it displays a navigation bar with 'namespace: demo > OWASP-Benchmark/BenchmarkJava > master'. Below the navigation are buttons for 'Rescan Project', 'Raw Data', and 'Export SBOM'. The main interface has tabs for 'FINDINGS 3.2K', 'INVENTORY', 'REMEDiations' (which is selected), 'PR RUNS 0', 'SCAN HISTORY 17', 'SECURITY REVIEW', 'ANALYTICS', and 'SETTINGS'. The 'REMEDiations' tab shows a search bar and a table of dependencies. The table includes columns for 'Dependency', 'Affected Package', 'Current Version', 'Endor Patch Available', and 'Fixed Vulnerabilities'. Remediation options are listed as 'Patch' (5.3.39-endor-2024-09-17, Low risk), 'Upgrade' (6.2.8, High risk), and 'Upgrade' (6.2.4, High risk). The right side of the interface shows a detailed view of a specific dependency: org.springframework:spring-webmvc. It includes sections for 'Patch Instructions', 'OVERVIEW' (Remediation Risk: Low, Findings Fixed: 2, Version Age: 10m ago), 'TESTS RUN' (Line of Code Changes: +272-10, Last Scan: N/A), 'Remediation Risk Drivers' (Potential Conflicts: 6, Breaking Changes: 0, Confidence: 0), 'Details' (Project: OWASP-Benchmark/BenchmarkJava, Package: org.springframework), and 'Patch Attestations' (Build Logs, Test Logs, Deploy Logs, Reproducible Build). A 'Fixed Findings' section shows 2 findings, including one related to a Path traversal vulnerability in functional web frameworks.



Go Beyond Finding

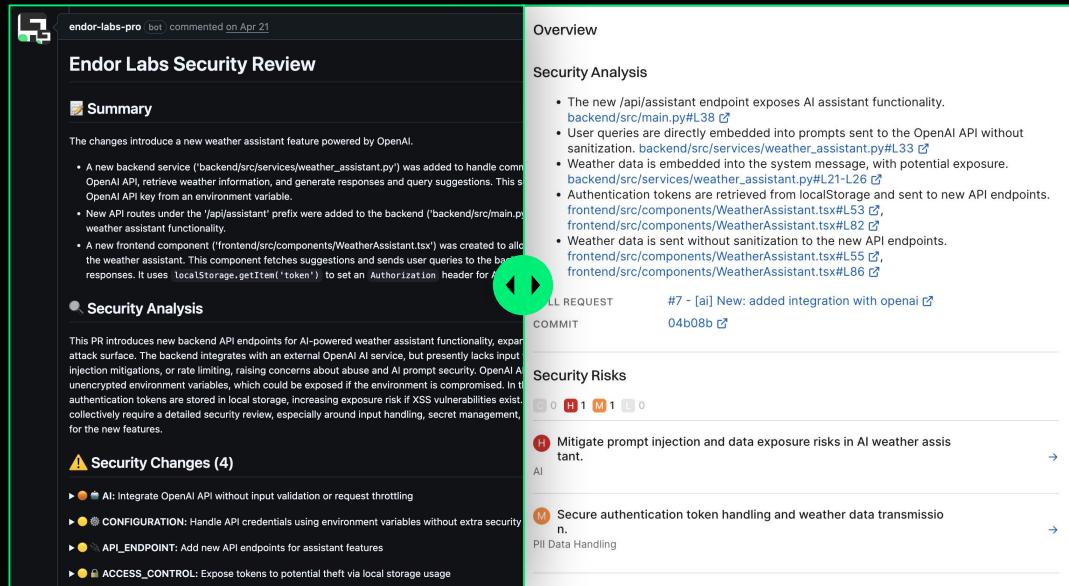
Endor Patches

Upgrade Impact Analysis

SAST Autofix*

Container Remediation*

Prepare For The Revolution With AppSec Copilot & AI Security Code Reviews



The screenshot shows a GitHub pull request for 'Endor Labs Security Review'. The pull request has been commented on by 'endor-labs-pro bot' on April 21. The 'Overview' section includes a 'Security Analysis' section with the following bullet points:

- The new /api/assistant endpoint exposes AI assistant functionality. [backend/src/main.py#L38](#)
- User queries are directly embedded into prompts sent to the OpenAI API without sanitization. [backend/src/services/weather_assistant.py#L33](#)
- Weather data is embedded into the system message, with potential exposure. [backend/src/services/weather_assistant.py#L21-L26](#)
- Authentication tokens are retrieved from localStorage and sent to new API endpoints. [frontend/src/components/WeatherAssistant.tsx#L53](#), [frontend/src/components/WeatherAssistant.tsx#L82](#)
- Weather data is sent without sanitization to the new API endpoints. [frontend/src/components/WeatherAssistant.tsx#L55](#), [frontend/src/components/WeatherAssistant.tsx#L86](#)

The 'Security Risks' section lists two items:

- H** Mitigate prompt injection and data exposure risks in AI weather assistant. [AI](#)
- M** Secure authentication token handling and weather data transmission. [PII Data Handling](#)

The pull request has 0 reviews, 1 approval, 1 comment, and 0 changes. The commit hash is 04b08b.



Secure AI Code and Models

AI Security Code Review

Security pair programmer via MCP

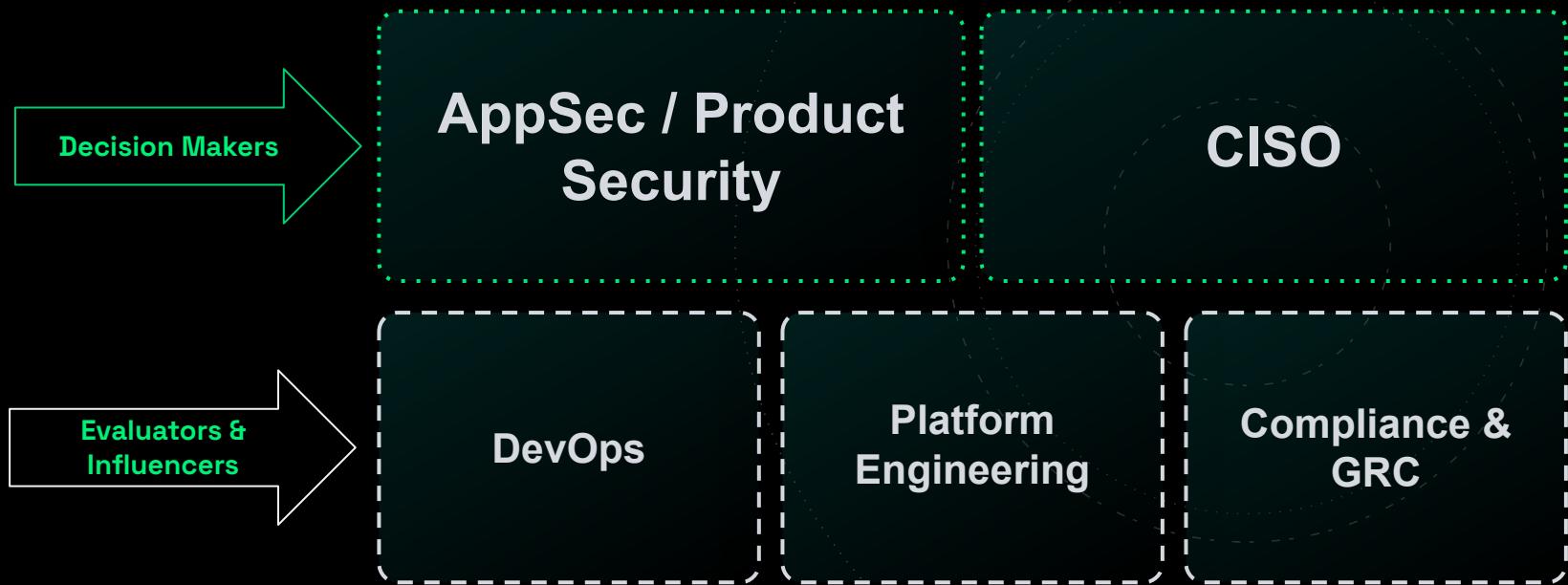
Model Discovery & Governance



Secure ~~open source~~ everything
your code depends on

Questions?

ENDOR LABS Target Audience



ENDOR LABS Opportunity Signals

- My Application Security Testing tools are **generating too much noise** and/or are **too expensive** for the value they provide.
- My **engineering teams won't fix vulnerabilities fast** enough.
- I'm concerned about emerging software supply chain risks like **malicious packages** or **LLM models**.
- I'm concerned about **AI generated code** - how will I keep up with the volume, and review code for security issues fast enough?

ENDOR LABS strengths in AI & AppSec

Prioritization

Endor Labs builds a graph of your code to **pinpoint risk at every layer** even in transitive dependencies.

Remediation

Endor Labs uses context about your code so **fixes don't break your applications**.

Automation

Endor Labs helps you adopt AI to build **dev-friendly workflows** and automate **traditionally manual efforts** like security code reviews.