



ENDOR LABS

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**

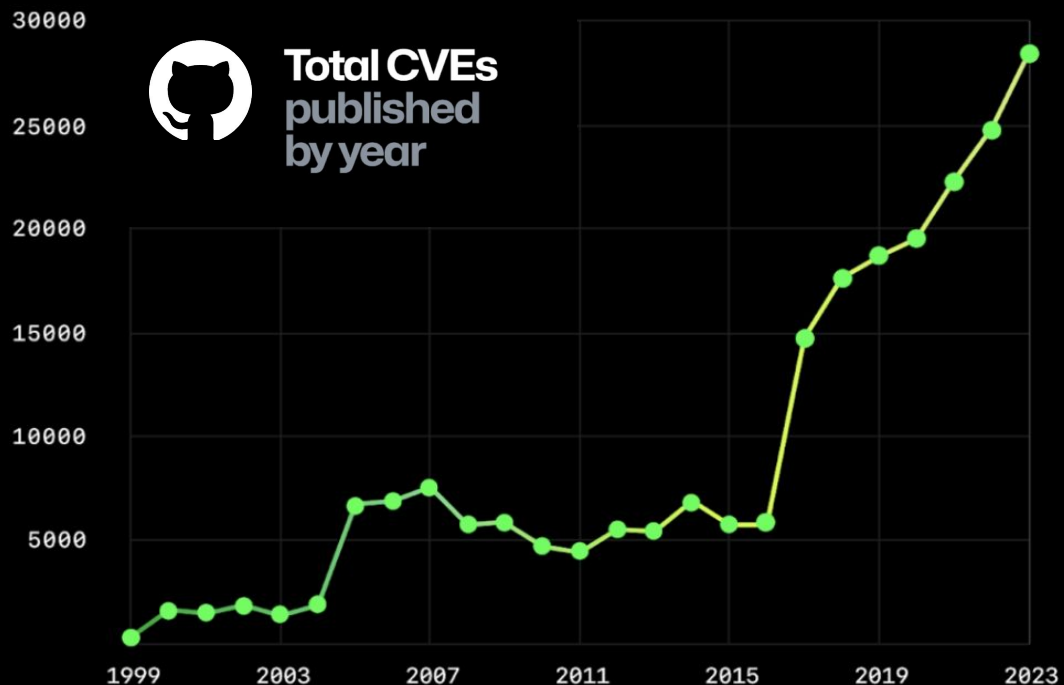
According to the research paper
"BAXBENCH: Can LLMs Generate
Correct and Secure Backends?"

42%

**of breaches exploit
web applications**

According to Verizon DBIR and
Mandiant's M-Trends

The Familiar Problems are Made Worse



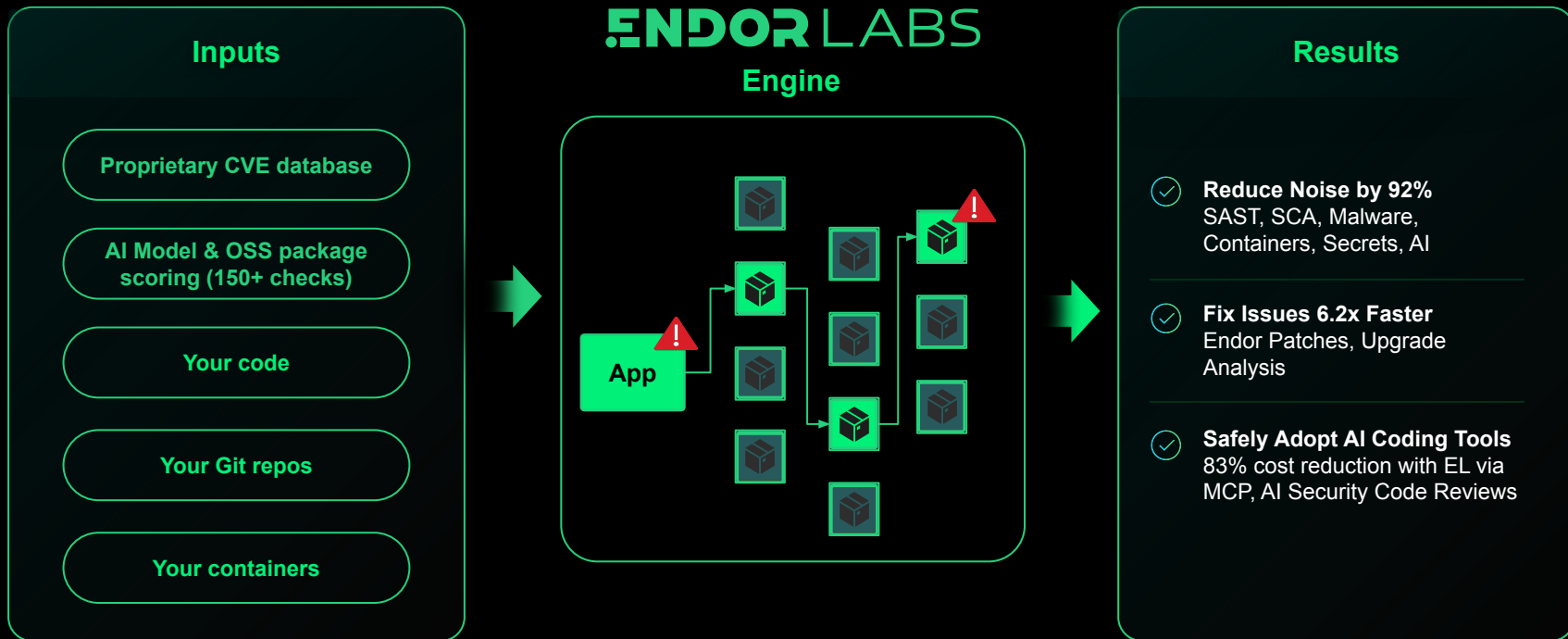
**Too Much
Noise**

**Too Many
Tools**

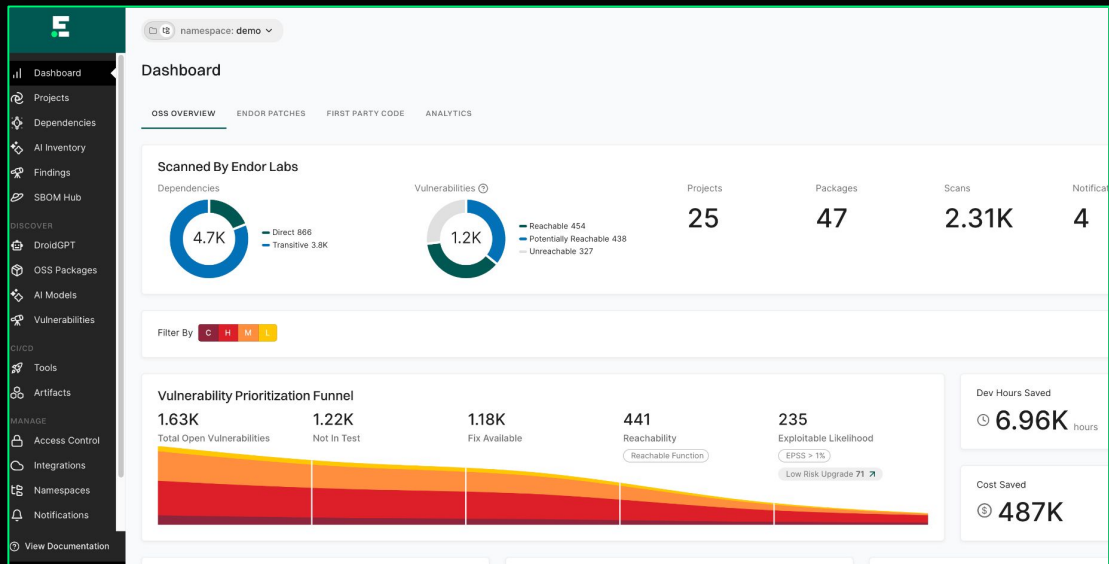
**Too Long to
Remediate**

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



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 displays the OWASP-Benchmark/BenchmarkJava remediation interface. The main table lists dependencies with their current versions and remediation options. The detailed view for `org.springframework:spring-webmvc` shows three remediation options: a patch, an upgrade to 6.2.8, and an upgrade to 6.2.4, each with associated risk levels and fixed findings.

Dependency	Affected Package	Current Version	Endor Patch Available	Fixed Vulnerabilities
✓ org.springframework:spring-jdbc	✓ org.owasp.benchmark:pom.xml	5.3.39	✗	0 0 0 0 1
✓ org.hibernate:hibernate-core	✓ org.owasp.benchmark:pom.xml	3.6.10.Final	✗	1 1 2 0 0
✓ org.springframework:spring-webmvc	✓ org.owasp.benchmark:pom.xml	5.3.39	✓	0 0 2 0 1

Remediation Options	Remediation Risk	Fixed Findings
Patch → 5.3.39-endor-2024-09-17 Recommended	Low	0 0 2 0 0
Upgrade → 6.2.8 Recommended	High	0 0 2 0 1
Upgrade → 6.2.4	High	0 0 2 0 1



Go Beyond Finding

Endor Patches


Upgrade Impact Analysis

SAST Autofix*

Container Remediation*

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





endor-labs-pro bot commented on Apr 21

Endor Labs Security Review

Summary

The changes introduce a new weather assistant feature powered by OpenAI.

- A new backend service ('backend/src/services/weather_assistant.py') was added to handle OpenAI API, retrieve weather information, and generate responses and query suggestions. This service uses an OpenAI API key from an environment variable.
- New API routes under the '/api/assistant/' prefix were added to the backend ('backend/src/main.py') to handle weather assistant functionality.
- A new frontend component ('frontend/src/components/WeatherAssistant.tsx') was created to allow the weather assistant. This component fetches suggestions and sends user queries to the backend. It uses 'localStorage.getItem('token')' to set an 'Authorization' header for API requests.

Security Analysis

This PR introduces new backend API endpoints for AI-powered weather assistant functionality, expanding the attack surface. The backend integrates with an external OpenAI AI service, but presently lacks input injection mitigations, or rate limiting, raising concerns about abuse and AI prompt security. OpenAI AI unencrypted environment variables, which could be exposed if the environment is compromised. In addition, authentication tokens are stored in local storage, increasing exposure risk if XSS vulnerabilities exist. Collectively, these changes require a detailed security review, especially around input handling, secret management, and the new features.

Security Changes (4)

- AI: Integrate OpenAI API without input validation or request throttling
- CONFIGURATION: Handle API credentials using environment variables without extra security
- API_ENDPOINT: Add new API endpoints for assistant features
- ACCESS_CONTROL: Expose tokens to potential theft via local storage usage

Overview

Security Analysis

- 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](#)

ALL REQUEST

#7 - [ai] New: added integration with openai

COMMIT

04b08b

Security Risks

0

1

1

0

- H

Mitigate prompt injection and data exposure risks in AI weather assistant.
- M

Secure authentication token handling and weather data transmission.

AI

PII Data Handling

Secure AI Code and Models

AI Security Code Review

Security pair programmer via MCP

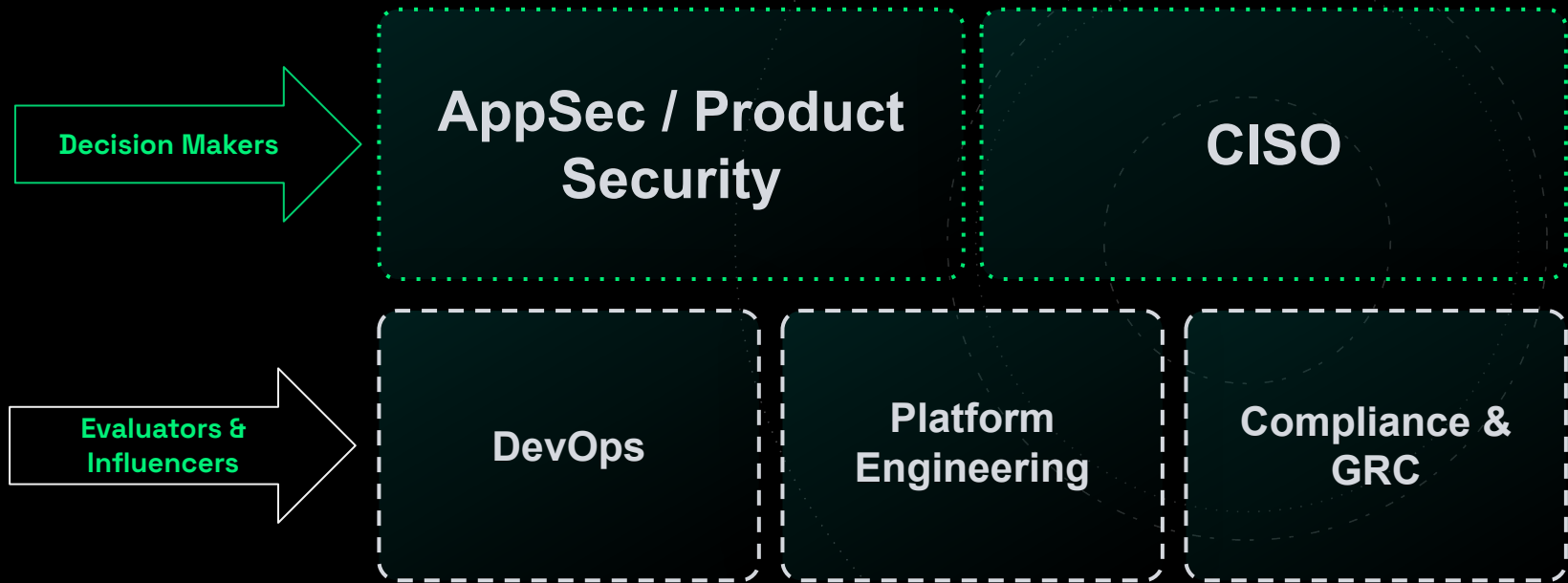
Model Discovery & Governance



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.