

Spring 2025 Release

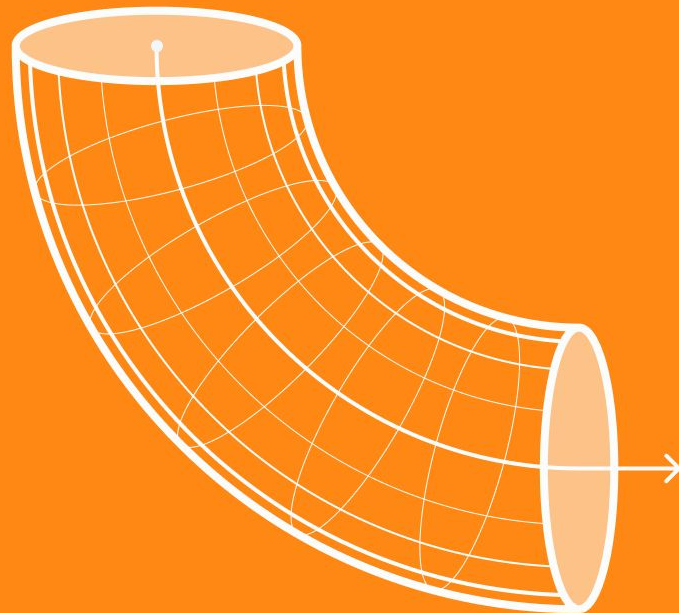


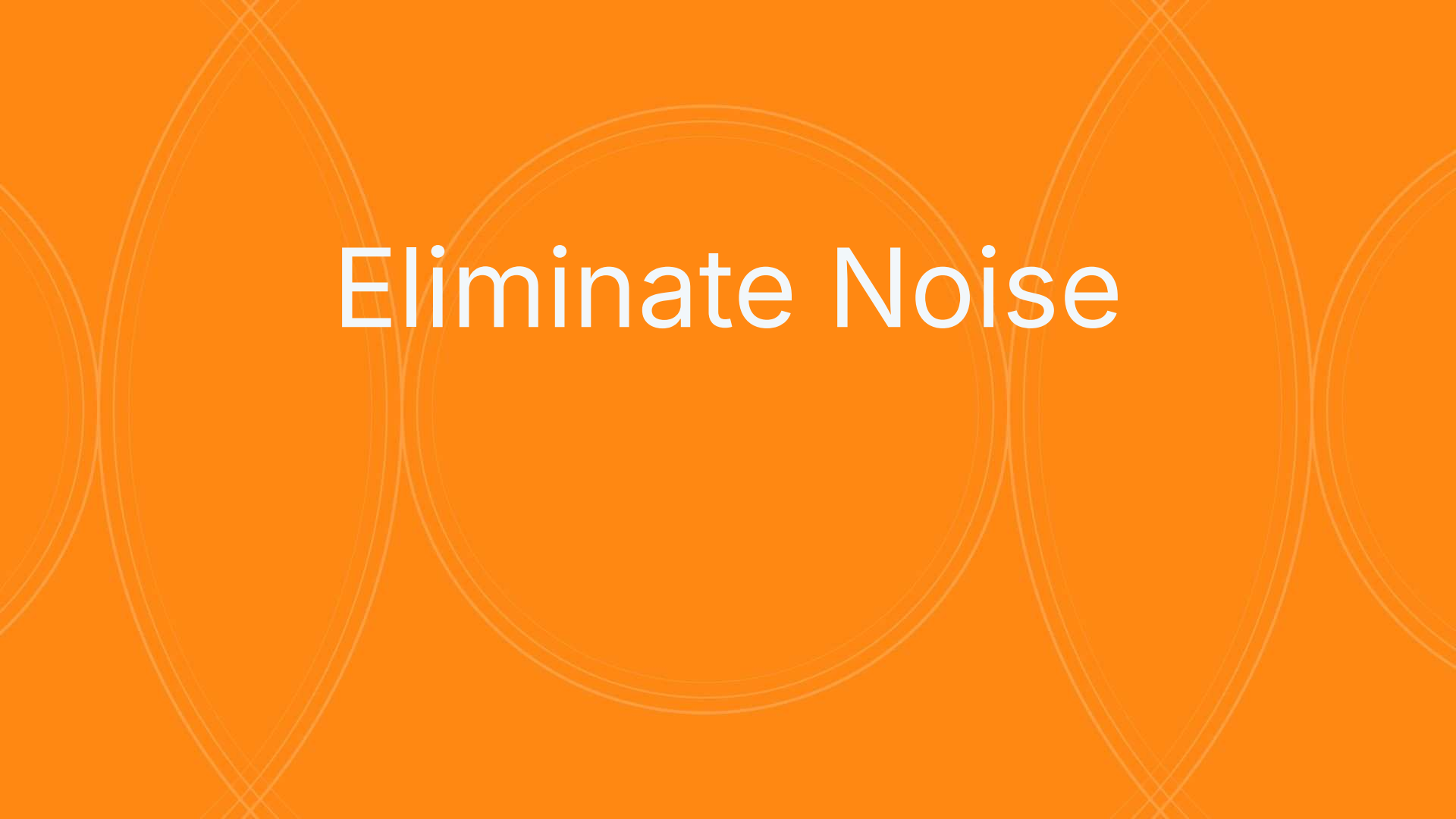
Table of Contents

1. [Overview](#)
2. [Eliminate Noise](#)
3. [Put AppSec on Autopilot](#)
4. [Operationalize and Scale](#)
5. [Maximize Coverage](#)
6. [Resources](#)

What's New for Spring 2025

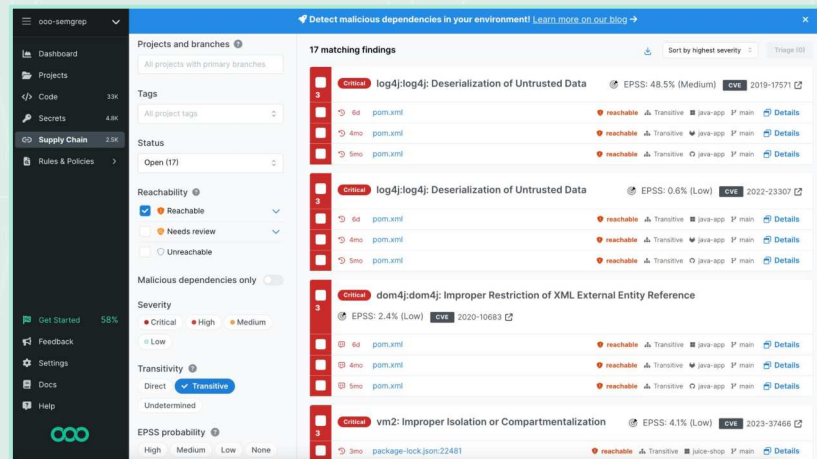
Our biggest update this spring is the public beta of Assistant Memories, which identifies 85% of false positives with no manual customization or tuning. With Assistant Memories, security engineers never have to triage the same issue twice, as Semgrep Assistant learns from prior triage decisions to eliminate contextual false positives.

We've also doubled down on enterprise-ready scanning to make it easier to secure large codebases. Semgrep Managed Scanning is now generally available across GitHub, GitLab, Bitbucket, and Azure DevOps – so teams can roll out scanning across every repo, team, and workflow. Plus, new support for lockfileless scanning expands coverage even while lockfiles are missing.

The background is a solid orange color with a pattern of overlapping, thin white circles. The circles are arranged in a way that they create a series of larger, irregular shapes, resembling a chain of bubbles or a stylized floral pattern. The text "Eliminate Noise" is centered in the middle of the image in a white, sans-serif font.

Eliminate Noise

Transitive Reachability



What it is

Semgrep now supports transitive reachability analysis for JavaScript projects (private beta). Transitive reachability extends analysis beyond direct dependencies to transitive dependencies. Semgrep flags vulnerabilities that are unreachable, helping teams reduce noise.

Why it matters

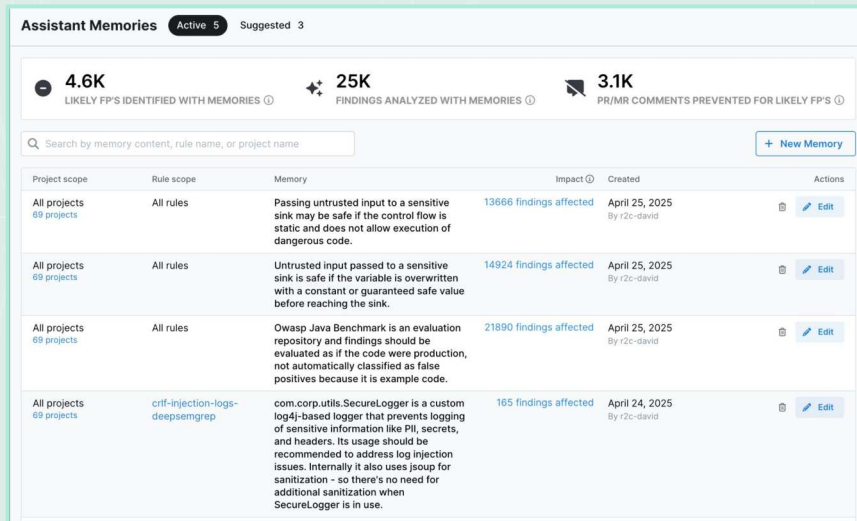
Not all known vulnerabilities pose a risk. By identifying when a transitive dependency is not used, Semgrep helps security teams reduce noise and prevent unnecessary work.

Learn more

[Transitive dependencies and reachability analysis](#)

PRIVATE BETA - COMING SOON

Assistant Memories



Assistant Memories Active: 5 Suggested: 3

4.6K LIKELY FPs IDENTIFIED WITH MEMORIES ⓘ

25K FINDINGS ANALYZED WITH MEMORIES ⓘ

3.1K PR/MR COMMENTS PREVENTED FOR LIKELY FPs ⓘ

Search by memory content, rule name, or project name [+ New Memory](#)

Project scope	Rule scope	Memory	Impact ⓘ	Created	Actions
All projects 69 projects	All rules	Passing untrusted input to a sensitive sink may be safe if the control flow is static and does not allow execution of dangerous code.	13666 findings affected	April 25, 2025 By r2c-david	Edit
All projects 69 projects	All rules	Untrusted input passed to a sensitive sink is safe if the variable is overwritten with a constant or guaranteed safe value before reaching the sink.	14924 findings affected	April 25, 2025 By r2c-david	Edit
All projects 69 projects	All rules	Owasp Java Benchmark is an evaluation repository and findings should be evaluated as if the code were production, not automatically classified as false positives because it is example code.	21890 findings affected	April 25, 2025 By r2c-david	Edit
All projects 69 projects	crif-injection-logs-deepsemgrep	com.corp.utils.SecureLogger is a custom log4j-based logger that prevents logging of sensitive information like PII, secrets, and headers. Its usage should be recommended to address log injection issues. Internally it also uses jsoup for sanitization - so there's no need for additional sanitization when SecureLogger is in use.	165 findings affected	April 24, 2025 By r2c-david	Edit

PUBLIC BETA

What it is

With Assistant Memories, Semgrep Assistant now has the ability to learn from triage notes, developer feedback, and explicit instructions in human language to filter out *even more* false positives.

Why it matters

For customers who use Memories, Assistant identifies **more than 85%** of all false positives without any human intervention.

Learn more

[Why AI-Powered Memories are the Future of SAST](#)

Cross-File Dataflow Traces

What it is

Now you can visualize how data moves across multiple files with in-app trace snippets.

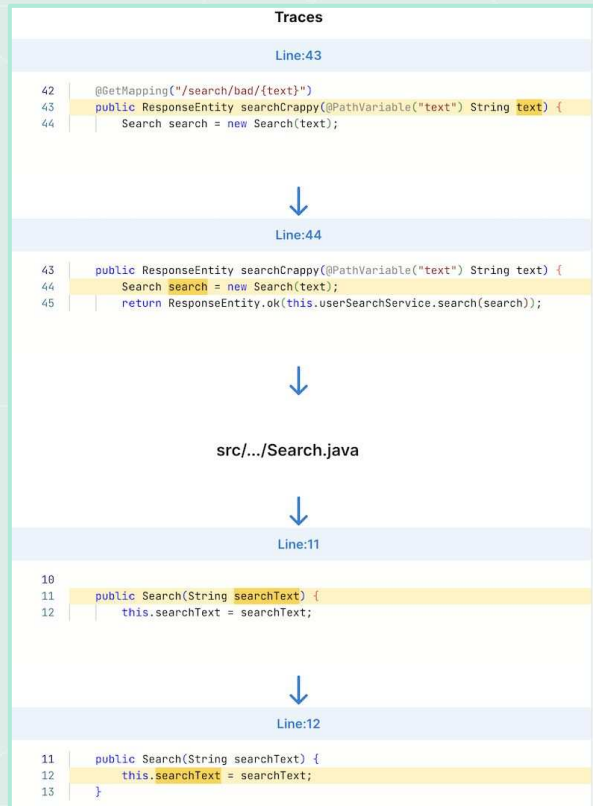
Why it matters

This improvement makes triage faster & easier, providing users the complete code context, all in one place.

Learn more

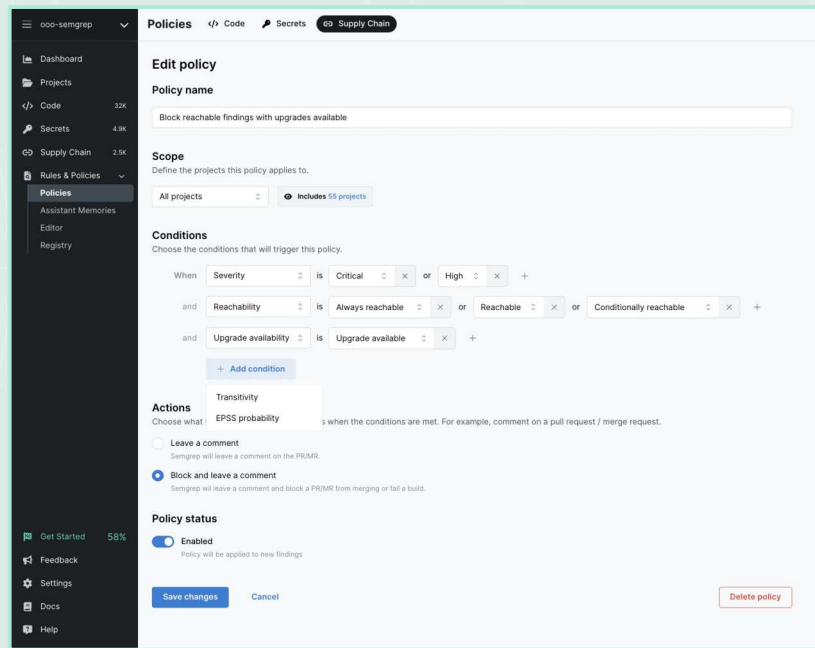
[In-app code snippets are now supported for cross-file dataflow traces](#)

GA



Put AppSec on Autopilot

Supply Chain Policies



What it is

Enforce precise configuration of supply chain policies based on criteria such as reachability, severity, upgrade availability, transitivity, Exploit Prediction Scoring System (EPSS) scores, and more.

Why it matters

You get smarter, super targeted controls that help your team fix what matters, ignore the noise, and stay ahead of supply chain threats.

Learn more

[Manage policies](#)

Suggested Memories

Assistant Memories Active 5 Suggested 3					
Search by memory content, rule name, or project name					
Project scope	Rule scope	Memory	Findings	Referenced triage comment(s)	Actions
ooo-gh/supply-chain-second-demo	tainted-pickle-flask	Treat data returned by <code>utils.clean()</code> as sanitized. If tainted request data is processed by <code>utils.clean()</code> before reaching <code>pickle.loads()</code> , the insecure-deserialization rule should not alert.	1 would be affected 1 in scope	"this data has been sanitized by <code>utils.clean</code> " semgrep-jamie on May 13, 2025	  Activate
All projects 69 projects	ssrf-deepsemgrep	When domain validation or input sanitization is done before building a request, the SSRF rule can flag a false positive. Ensure that strict controls prevent untrusted requests from reaching unauthorized endpoints.	606 would be affected 2846 in scope	"We're validating the called domain from the user request so this SSRF is a false positive." r2c-nitin on February 10, 2025	  Activate

What it is

Semgrep Assistant can now suggest Memories based on triage notes and developer feedback. Users can easily see how many findings are in scope for each suggested memory, and view the impact of turning them on.

Why it matters

Manual triage is costly and error-prone. Assistant memories solve this issue, letting security teams use human language to capture organizational context, reducing false positives.

This turns manual triage from grunt work into a strategic, compounding investment that permanently reduces noise for developers.

PUBLIC BETA

Assistant for Bitbucket and Azure DevOps



What it is

Semgrep Assistant now supports Bitbucket and Azure DevOps (ADO), in addition to GitHub and GitLab. Developers using Bitbucket and ADO now receive remediation guidance, autofixes, and explanations natively in PR comments.

Why it matters

By supporting additional SCMs, Semgrep removes friction for teams using diverse repositories, ensures faster time-to-remediation, and accelerates adoption across engineering teams that aren't on GitHub.

Learn more

[Bitbucket docs](#)

[Azure DevOps docs](#)

LLM Selection / BYO API Key

Allowed AI providers ×

Choose the AI providers Assistant can use. If you enable multiple, Semgrep will choose the best model for the task. Enabling multiple models may improve Assistant's performance overall.

☒ **OpenAI Providers**

- ☒ Semgrep's OpenAI API key
Zero Data Retention enabled – OpenAI does not log any inputs or outputs
- ☐ Your OpenAI API key
- ☐ Azure OpenAI

☒ **AWS Bedrock**

☐ **Google Gemini**

☐ **xAI**

Save

What it is

Semgrep Assistant now supports multiple LLM providers including OpenAI, AWS Bedrock, Google Gemini, and xAI. Users can also bring their own API key to leverage their own relationships with major model providers.

Why it matters

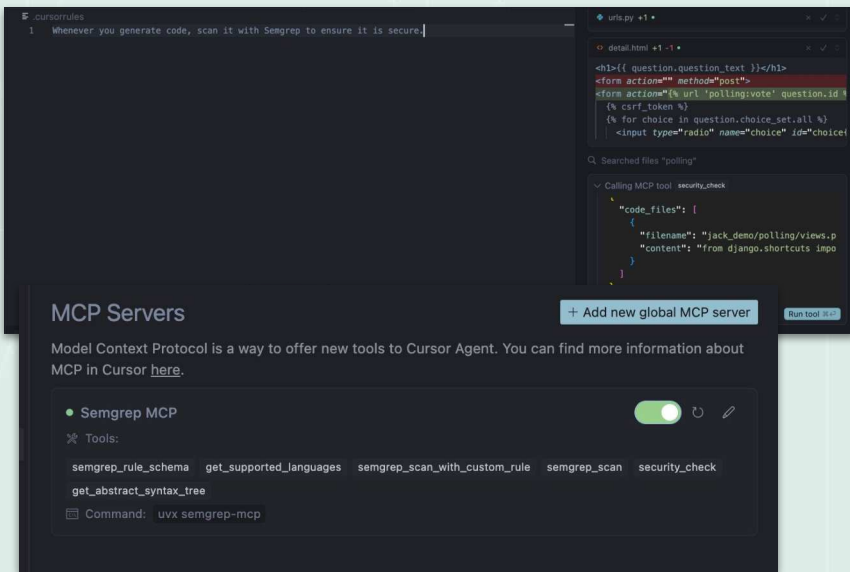
Every organization has different privacy, cost, and performance requirements when it comes to LLMs. With multi-model support and BYO API keys, teams can choose the model that best aligns with their policies and priorities—whether that's compliance, speed, cost-efficiency, or output quality.

This also lets users leverage any existing data processing or privacy agreements with vendors.

Learn more

- [Documentation](#)

Semgrep MCP Server



What it is

The Semgrep MCP Server turns Semgrep into a built-in security reflex for LLMs—letting them scan and fix their own code *as they generate it*.

Why it matters

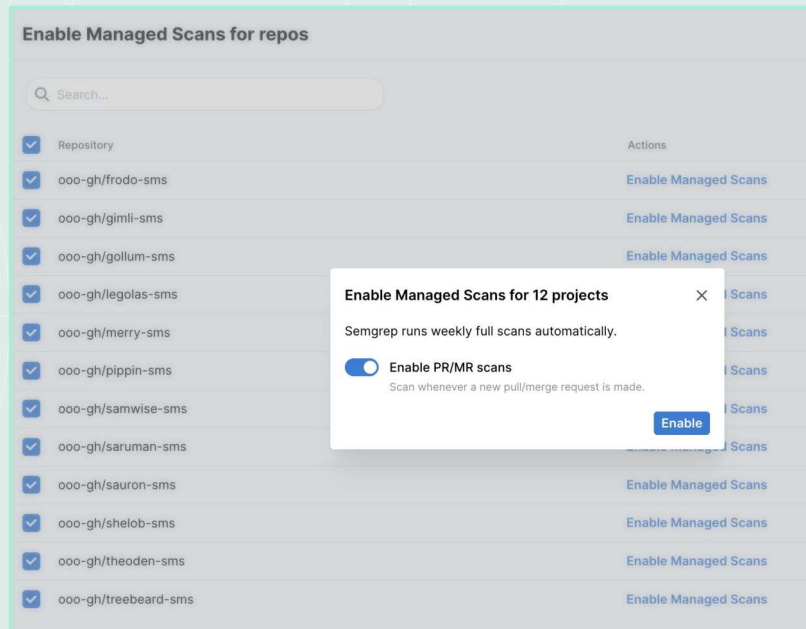
The amount of AI-written code has already exploded and will increase exponentially. Semgrep's MCP server let's any LLM scan generated code for security issues, and use the context Semgrep provides to accurately fix issues.

Learn more

[Giving AppSec a Seat at the Vibe Coding Table](https://semgrep.dev/resources/whats-new)

Operationalize and Scale

Enterprise-Ready Scanning



GA

What it is

Semgrep Managed Scanning syncs, onboards, and scans all your repositories — automatically, saving you time from managing CI/CD pipelines. Whether you have a large codebase or complex monoliths, Managed Scanning will automatically scale to ensure complete scanning coverage across any repository in a timely manner.

Why it matters

These improvements accelerate onboarding, ensure faster, more reliable results, and improve scan completion rates for high-complexity environments.

Learn more

[Rapidly deploy code scans across your organization with Semgrep managed scanning](#)

Scan Without Lockfiles

The screenshot displays the Semgrep AppSec Platform interface. On the left, a sidebar shows a list of projects: 'test_ptt' (16 dependencies), 'Semgrep-Demo/js-app' (2649 dependencies), and 'Semgrep-Demo/python-app' (67 dependencies). The 'test_ptt' project is selected, showing a table of dependencies from 'package-lock.json'.

Dependency	Dependency type
@storybook/react 7.6.17	Direct
cypress 13.6.6	Direct
lodash 4.17.21	Transitive
prop-types 15.7.2	Direct

Below the table, there are links to 'See all 16 dependencies' and 'See all 67 dependencies' for other projects.

On the right, a detailed view of the 'test_ptt / package-lock.json' is shown. It displays a 'Dependency path' tree starting from 'root packages'. The path includes 'storybook/react 7.6.17' (Direct), 'packagexyz 3.4.1' (Transitive), 'packagejkl 2.3.4' (Transitive), 'lodash 4.17.21' (Transitive, marked as VULNERABLE), 'packageabc 2.3.7' (Transitive), 'lodash 4.17.21' (Transitive, marked as VULNERABLE), 'cypress 13.6.5' (Direct), 'packagedef 1.5.2' (Transitive), and 'lodash 4.17.21' (Transitive, marked as VULNERABLE).

What it is

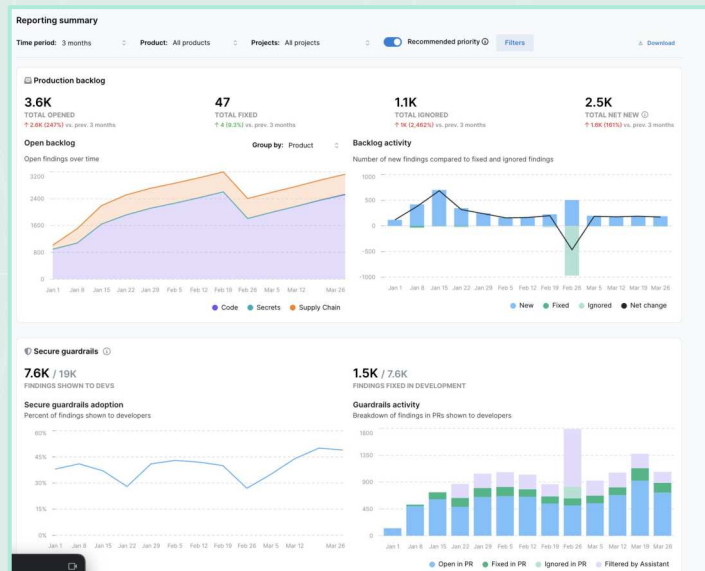
Now users can scan projects even when lockfiles are missing or difficult to generate — common in compiled languages such as Java, C#, and Kotlin.

Why it matters

Provides accuracy regardless of how your development team manages dependencies. Reduces setup friction for AppSec teams, boosts coverage, reduces blind spots, and enables security across monorepos and diverse dependency management practices.

PRIVATE BETA

Clickable Charts



PUBLIC BETA

What it is

Revamped security dashboards that turn Semgrep scan data into interactive charts, trends, and reports tailored for developers, AppSec teams, and CISOs. Provides an overview of your organization's security posture.

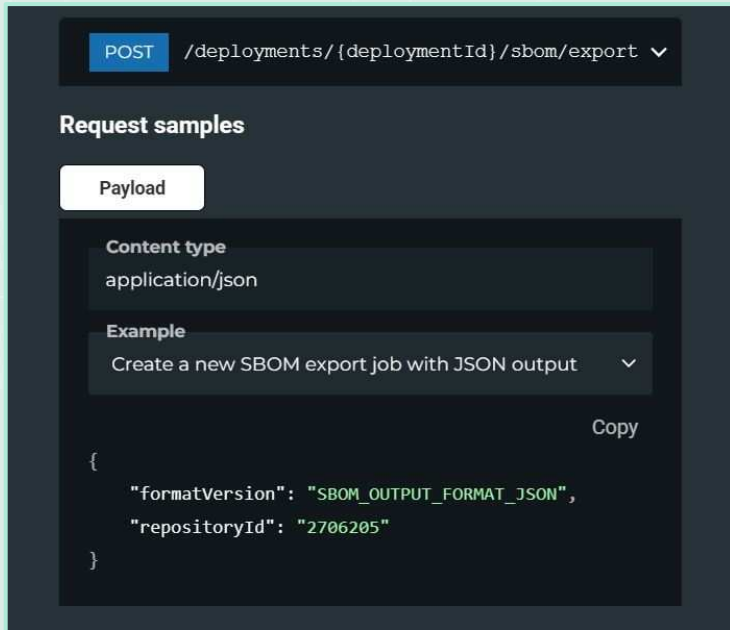
Why it matters

Enables faster triage, clearer risk visibility, and stronger alignment across teams—saving time, improving communication, proving ROI, and strengthening overall security posture.

Learn more

[Revamped reporting Dashboards docs](#)

SBOM Export API



GA

What it is

Allows users to programmatically retrieve a Software Bill of Materials (SBOM) for their scanned projects using public, documented endpoints.

Why it matters

Enables easier integration with compliance tools and workflows, replaces reliance on internal APIs.

Learn more

[Semgrep API - SBOM Documentation](#)
[Generate a Software Bill of Materials](#)



Maximize Coverage

Expanded JavaScript & TypeScript Analysis

Benchmark	Value
True positive rate (before AI processing) for latest <code>p/default</code> ruleset	63%
Lines of code scanned	~8 million
Repositories scanned	153
Findings triaged to date	~600

GA

What it is

Improved detection capabilities for JavaScript and Typescript, including engine-level dataflow analysis for 50+ frameworks and libraries, including Express, NestJS, React, and Angular.

Why it matters

You'll get broader and deeper detection for modern JavaScript and TypeScript codebases.

Learn more

[Beyond Benchmarks: How Semgrep Redefines Javascript Security](#)

Shadow AI Ruleset

What it is

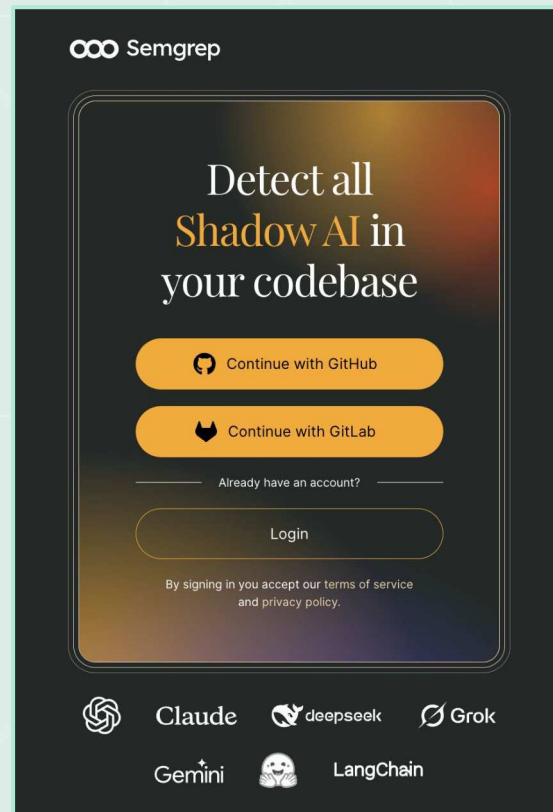
A new ruleset that detects unauthorized use of AI or LLM libraries. This includes API calls, such as api.openapi.com, and libraries in code such as langchain and transformers.

Why it matters

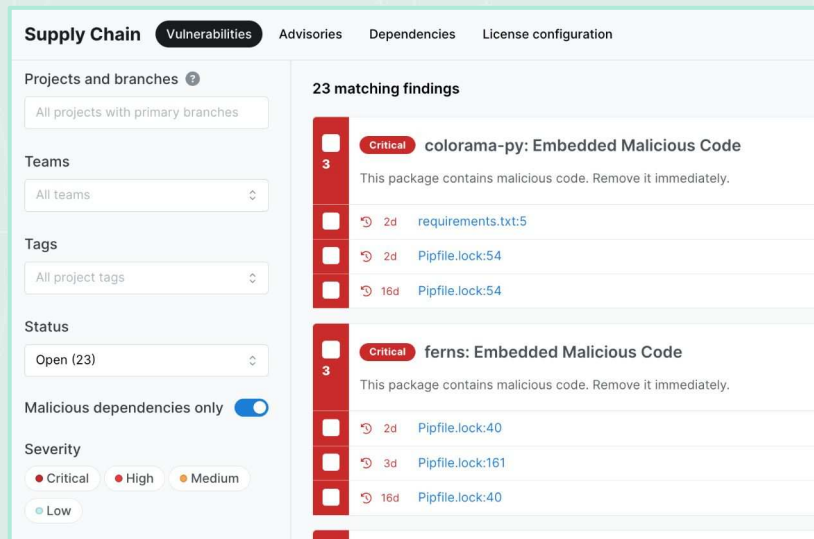
Unauthorized AI usage can expose sensitive data, violate compliance standards, and increase risk of prompt injection and leaked credentials. Now you can catch risky LLM usage before your code ships.

Learn more

[Semgrep Shadow AI](#)



Malicious Dependency Detection



PUBLIC BETA

What it is

Malicious dependencies are dangerous packages, or dangerous versions of packages, that are designed to compromise systems.

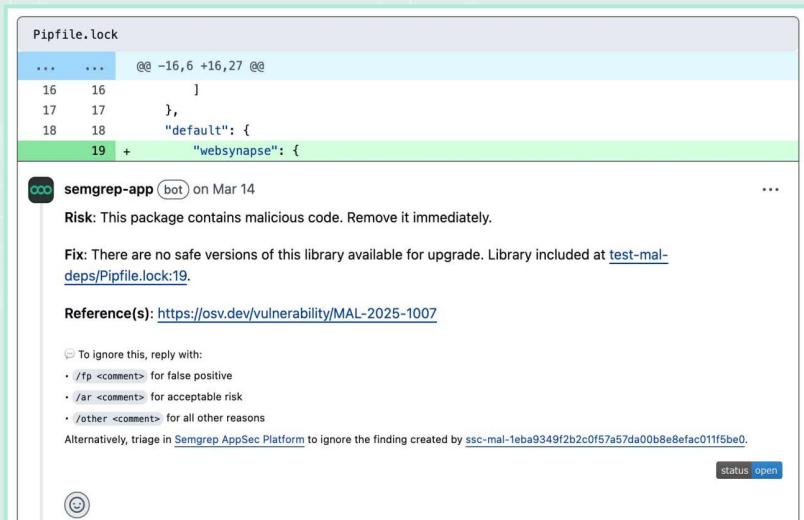
Why it matters

Now teams can identify and block known malicious packages – such as those involved in typosquatting or supply chain attacks – across ecosystems like npm, PyPI, RubyGems, Cargo, Go, and NuGet.

Learn more

[Detect and remove malicious dependencies](#)
[Beyond vulnerabilities: Detect malicious dependencies in your supply chain](#)

PR Warnings for Malicious Packages



What it is

Semgrep will now comment directly on pull requests or merge requests warning users that they may be adding malicious dependencies.

Why it matters

Dev teams get real-time feedback to stop risky dependencies before they're merged.

Learn more

[Detect and remove malicious dependencies](#)

Generic Secrets Detection

☐	password-string
2	Generic Secret (AI)
☐	 1mo <code>src/main/resources/application-postgresql.properties:4</code>
☐	 1mo <code>routes/login.ts:66</code>

GA

What it is

Combines rules and LLM-powered filtering to help detect generic secrets accurately.

Why it matters

This allows users to cast a wider net to catch more secrets without overwhelming teams with false positives.

Learn more

[Semgrep Secrets overview](#)

Wiz Integration

The screenshot displays the Wiz Security Graph interface. The top section shows a list of findings from Semgrep. The bottom section provides a detailed view of a specific finding, 'CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')'.

Finding	Resource	File Path	Status	Severity	First seen at	External Source
CWE-321: Use of Hard-coded Cryptographic Key	Semgrep-Demo/juice-shop/ma... Repository Branch	frontend/src/app/last-login-ip/last-login-ip.component.spec.ts	Unresolved	High	February 12, 2025 at 2:50 AM	Semgrep Pro Engine
CWE-89: Improper Neutralization of Special...	Semgrep-Demo/juice-shop/ma... Repository Branch	data/static/codefixes/unionSqlInjecti	Unresolved	High	February 12, 2025 at 2:50 AM	Semgrep Pro Engine
CWE-943: Improper Neutralization of Special...	Semgrep-Demo/juice-shop/ma... Repository Branch	routes/showProductReviews.ts	Unresolved	High	February 12, 2025 at 2:50 AM	Semgrep Pro Engine
CWE-89: Improper Neutralization of Special...	Semgrep-Demo/juice-shop/ma... Repository Branch	data/static/codefixes/dbSchemaChal	Unresolved	High	February 12, 2025 at 2:50 AM	Semgrep Pro Engine
CWE-943: Improper Neutralization of Special...	Semgrep-Demo/juice-shop/ma... Repository Branch	routes/orderHistory.ts	Unresolved	High	February 12, 2025 at 2:50 AM	Semgrep Pro Engine
CWE-94: Improper Control of	Semgrep-Demo/juice-shop/ma... Repository Branch	routes/feedback.ts	Unresolved	High	February 12, 2025 at 2:50 AM	Semgrep Pro Engine

CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
Semgrep Pro Engine Finding

Ignore Comment Support

Overview Remediation

The vulnerability was discovered in file `data/static/codefixes/unionSqlInjectionChallenge_3.ts`, lines `10-10`, identified by commit hash `766f109e0fa5db79cd7d4f135d3e3c1a60197e64`.

Detected user input used to manually construct a SQL string. This is usually bad practice because manual construction could accidentally result in a SQL injection. An attacker could use a SQL injection to steal or modify contents of the database. Instead, use a parameterized query which is available by default in most database engines. Alternatively, consider using an object-relational mapper (ORM) such as Sequelize which will protect your queries.

Source Code: [github.com](#)

NVD Severity: -

Version: -

External Source: Semgrep Pro Engine

Severity: **High**

Project: semgrep

Fixed Version: -

Data Source: [semgrep.dev](#)

Vendor Severity: -

Component name: -

Detection Method: External Network Scan

Status: **Unresolved**

First seen: Feb 12, 2025, 2:50 AM

Last seen: Feb 12, 2025, 2:50 AM

What it is

Semgrep integrates with Wiz by establishing a secure connection with Wiz's API endpoints, enabling you to prioritize vulnerabilities by correlating SAST findings with real-time cloud infrastructure and runtime data.

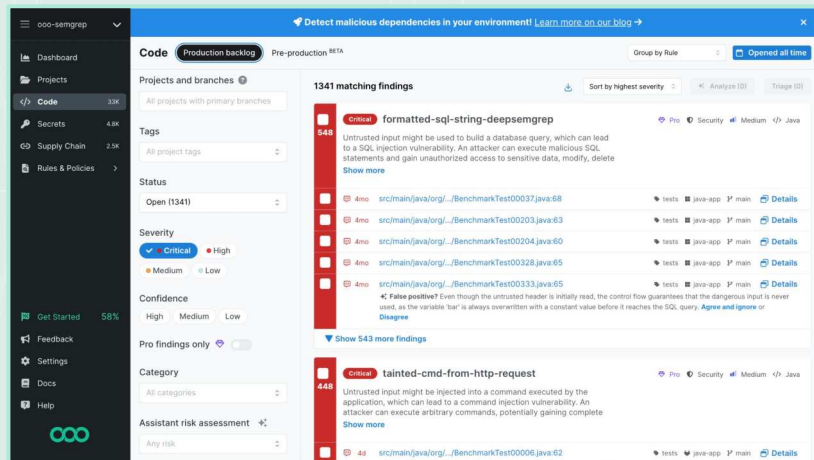
Why it matters

Now you can get a holistic view of your code and infrastructure security so that you can focus on what matters most.

Learn more

[View Semgrep findings in Wiz's Security Graph](#)

Critical Severity Classification



What it is

The Critical severity level is now available in Semgrep Code and Semgrep Secrets to denote the highest severity for both Semgrep Code and Semgrep Secrets findings.

Why it matters

Security teams can now prioritize the most urgent issues. Plus, they can identify high risk rules in the Semgrep Registry that generate Critical findings to help teams focus remediation where it counts most.

Learn more

[Announcing Critical Severity for Semgrep Code & Secrets](#)

Resources

Resources



Visit the quarterly release page

Get a detailed look at this quarter's latest innovations.

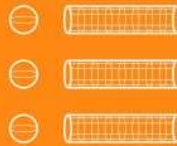
semgrep.dev/resources/whats-new



Discover the latest product updates

Stay informed about significant new features and enhancements.

semgrep.dev/products/product-updates/



Check out the release notes

Understand the full scope of changes in each release.

semgrep.dev/docs/release-notes



Learn AppSec with Semgrep Academy

Learn to create secure software with us.

academy.semgrep.dev