

Semgrep Application Security Platform

Protect your code with secure guardrails

Developers are incentivized to ship features, not to fix security issues. When facing noisy tools that produce false positives without actionable guidance on fixes, they prioritize feature velocity. The backlog grows, increasing security debt with every release, creating an AppSec doom loop.

About Semgrep

Semgrep makes it easy to fix critical vulnerabilities today, while its secure guardrails guide developers towards best practices that prevent vulnerabilities tomorrow. With the Semgrep AppSec Platform—including SAST, SCA, and secrets—developers spend less time dealing with recurring vulnerabilities, accelerating releases and reducing risk.

Why choose Semgrep?

Say goodbye to false positives

- Use dataflow reachability to reduce false positives by up to 98%
- Find issues unique to your application code with easily customized rules
- Use AI to triage findings and identify true and false positives

Eliminate developer friction

- Only show actionable findings in the developer workflow
- Give developers actionable guidance during code review
- Get AI-generated code remediation using Semgrep Assistant

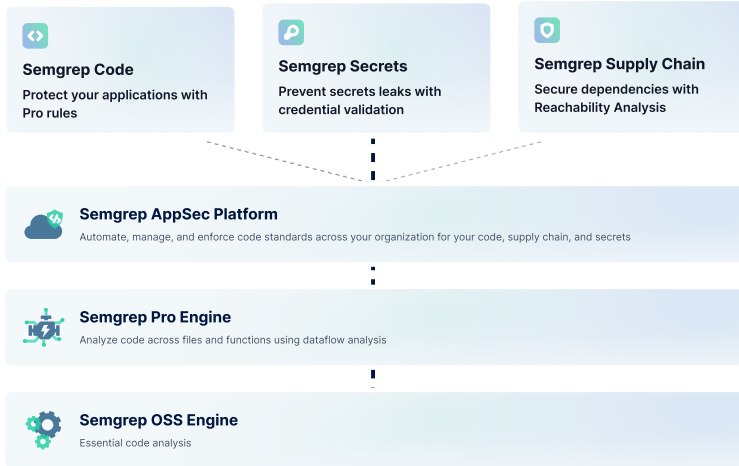
Prevent tomorrow's issues today

- Enforce organization-specific security practices
- Eliminate entire classes of vulnerabilities
- Prevent security issues by construction

Trusted by top security and engineering teams



"Semgrep Supply Chain has helped reduce the noise by 95%" - Lyft



Code, dependency, and secrets scanning in a single platform

- ✓ 30+ programming languages supported
- ✓ AI-powered triage, remediation, and rule generation with Semgrep Assistant
- ✓ Engage developers in their workflow: IDE, command-line, and code review
- ✓ Speed releases and reduce security debt with secure guardrails

Developer's view of Semgrep

semgrep-appsec-platform (bot) reviewed 2 weeks ago

```

src/assistant-fix-custom-message.java
11 + private final static Logger log = Logger.getLogger(Logger.GLOBAL_LOGGER_NAME);
12 + protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Serv
13 +     String param = request.getParameter("param");
14 +     log.info("foo"+param+"bar");
  
```

semgrep-appsec-platform (bot) 2 weeks ago

When data from an untrusted source is put into a logger and not neutralized correctly, an attacker could forge log entries or include malicious content. Please use the `Jsoup.clean()` function to sanitize data.

► View Dataflow Graph

Reply with `/semgrep ignore <reason>` or [triage in Semgrep AppSec Platform](#) to ignore the finding created by [crif-injection-logs-deepsemgrep-javaorg-copy](#).

semgrep-appsec-platform (bot) 2 weeks ago

Semgrep Assistant suggests the following fix: Sanitize the `param` variable using `Jsoup.clean()` before logging it.

► View step-by-step instructions

This code change should be a good starting point:

```

Suggested change
14 -     log.info("foo"+param+"bar");
14 +     String sanitizedParam = Jsoup.clean(param, Whitelist.none());
15 +     log.info("foo" + sanitizedParam + "bar");
  
```

Commit suggestion | Add suggestion to batch

>> [Get Started Now](#) | [Book a demo semgrep.dev/contact-us](#) <<