STACKHAWK

StackHawk & Snyk

Correlated dynamic and static security testing to prioritize and fix application and API security issues.

Development teams need security testing that keeps pace with modern deployment cycles. Traditional security tools slow down development with unclear findings, manual correlation work, and security feedback that arrives too late. Without a unified view from SAST to DAST, developers can't quickly identify which vulnerabilities are truly exploitable or where in their codebase they need to focus remediation efforts.

Accelerate the find and fix cycle with DAST + SAST

StackHawk and Snyk offer a developer-first security testing platform that correlates dynamic and static findings. Fully automated in CI/CD with developer-centric workflows built for modern apps, microservices, and APIs.

How it works

1. Prioritize findings, DAST and SAST testing works together to identify the high-priority, exploitable security issues in your code
2. Accelerate fixes, quickly identify where the issue exists in your codebase, down to a single line of code
3. Drive efficiency, eliminate context switching and understand application security issues at a glance

Interested in seeing StackHawk and Snyk in Flight?

Transform duplicate alerts into unified security intelligence that drives real risk reduction.

Book a Demo → View Integration Docs →

Measurable Impact

- **Reduce Noise**
  Alert developers only when code is merged and issues are found
- **Risk-Based Prioritization**
  Catch security issues before they are shipped to production
- **Faster Remediation**
  Equip developers with the information and tools to fix vulnerabilities quickly
- **Improved ROI**
  Comprehensive security coverage for modern apps, microservices, and APIs

Benefits for the Entire Team

- **AppSec Teams**
  Get comprehensive understanding of application security issues with correlated DAST and SAST findings that prioritize exploitable vulnerabilities.
- **Developers**
  Get the exact line of code where issues exist and fix vulnerabilities without context-switching, so you can get back to feature development faster.