BLACK DUCK®

GUIDE

ENHANCE VISIBILITY TO IMPROVE RISK MANAGEMENT FOR AI-DRIVEN DEVELOPMENT

You can't fix what you can't see. This simple maxim applies everywhere in life, including DevSecOps programs. Thus, end-to-end visibility across all stages of the software development life cycle (SDLC), from the developer desktop through build and CI pipelines, is critical for managing software risk. This is especially true as AI coding assistants (e.g., GitHub Copilot, Tabnine) generate large volumes of code at unprecedented speeds.

For every team, gaining a clear line of sight into issues as early and consistently as possible should be a principal goal. This will, however, require a concerted effort among key contributors.

- **Development teams.** Early visibility into security issues is critical. Developers should be made aware of issues as they code, declare dependencies, and check artifacts into source code management (SCM) systems and repositories. This helps them avoid late-stage rework, so they can move on to the next feature branch quickly.
- **DevOps teams.** Consistent visibility across the SDLC allows DevOps teams to understand the security status of artifacts passing through pipelines. This should include awareness of any direct and transitive dependencies that get resolved into the build, as well as the results of any functional and unit tests. Visibility for DevOps teams helps them make informed, incremental adjustments for timely and secure software releases, and ensure that they consistently achieve software shipping deadlines.
- **Application security (AppSec) teams.** Comprehensive visibility means that AppSec teams are aware of all development projects, understand development tools and build pipelines, and are alerted to the assets introduced through the software supply chain. Visibility for AppSec teams enables them to understand the attack surface and establish integrated, automated mechanisms to detect and resolve issues as quickly as possible.

Unfortunately, lack of visibility into projects, workflows, and security issues is a persistent problem that will only worsen with AI-enabled development, as the window for risk detection shrinks.

Two key factors impede visibility and prevent DevSecOps teams from improving risk management across their organization.

- **Distinct, high-velocity development activities.** Each project is composed of proprietary code, open source libraries, and third-party dependencies, all created using a variety of languages and frameworks across developers' preferred IDEs and build pipelines. Additionally, each application can be built for a variety of purposes (e.g., to run on-premises or in the cloud, or as monolithic, containerized, or serverless architectures) with each developer often focusing on a subset of technologies and functions. This creates silos and further obscures visibility. Without adequately accounting for these diverse tools and pipeline configurations, it can become difficult to catch issues consistently.
- **Separation of duties and organizational silos.** Developers are not solely responsible for application security, but they significantly impact an organization's security risk posture. Without a shared vision for DevSecOps, security gates can become painful impediments to development workflows. This can cause developers to establish secondary build pipelines and test environments outside the visibility of AppSec teams, precluding any ability to assess risk or recommend priority fixes.

DevSecOps requires a breakdown of silos and consistent visibility across projects, pipelines, and risks. To do this, AppSec processes should meet developers where they are, integrating and automating security tests directly into development tools and workflows. And AppSec teams need to treat AI coding assistants like human developers—fallible, task-oriented, yet essential supporters of security when empowered properly. Only then can each contributor accurately assess their risk posture and adjust workflows accordingly to become more secure and efficient over time.

AppSec teams need to treat AI coding assistants like human developers—fallible and task-oriented, yet essential supporters of security when empowered properly.

INCREASE VISIBILITY BY PUTTING "EYES" EVERYWHERE

The quickest way to expand visibility into software risk is to integrate static application security testing (SAST) and software composition analysis (SCA) at every stage of CI pipelines. This puts critical mechanisms for risk detection wherever it might enter a project and keeps prioritization and remediation close to development. This will

- Align risk awareness across security, development, and DevOps teams
- Ensure continuous improvement and reduce issue backlogs
- Prevent deployment delays
- Avoid compromising on risk tolerance to meet deadlines

Further, automating these tests based on pipeline triggers ensures that security matches the speed of AI-enabled development. Security tests become a natural and unavoidable part of DevOps workflows, providing real-time feedback to developers and reducing the time between code changes and security assessments. Issues detected can be triaged and prioritized for remediation based on predefined risk tolerance policies, customized by project, and refined to balance security, speed, and compliance.

Critically, integrated testing must help developers code securely and cross-check AI coding assistants' contributions. This is most effectively done using IDE plug-ins, SCM extensions, and flexible AppSec platforms, rather than disrupting developers' workflows.

Integration testing must help developers code securely and cross-check AI coding assistants' contributions.

IDE PLUG-INS: VISIBILITY AT THE SOURCE

Developers are often considered the first line of defense against security issues or software license conflicts. Allowing developers to see issues they (or their AI coding assistant) have introduced into project files can direct their attention to emergent risks. Moreover, providing remediation guidance within the IDE gives developers the knowledge to fix detected issues, and ideally, avoid introducing the issue again in the future.

Black Duck's **Code Sight™ IDE Plug-in** is a "security spellchecker" for developers that enhances real-time visibility and provides actionable remediation guidance. Code Sight is an effective and natural way to help developers code securely, providing capabilities including

- **Integrated SAST and SCA:** Code Sight detects flaws in code written by humans or AI code generators, and identifies vulnerable open source dependencies without requiring developers to change workflows or access separate tools.
- **On-demand scanning:** Developers can check their code at any point during the development process, and perform whole-project or single-file scans to optimize their efforts.
- **Automated scanning:** Scans can be run at specific intervals or be triggered by certain events, such as code commits, builds, or pull requests.
- **Balanced speed and depth:** Various scanning modes let developers choose the appropriate level of scrutiny based on the stage of development and criticality of the code.
- **Unified visibility for dev and AppSec:** By connecting Code Sight with other Black Duck solutions (e.g., Black Duck Polaris™ Platform), teams can see risks detected by pipeline scans outside the IDE and deliver policy-driven fix priorities to developers as well as the issues assigned to them for remediation. When scans are initiated within the IDE, associated project test results automatically appear in Polaris to ensure that AppSec teams are always aware of new development activity.

By incorporating Code Sight into the development environment, organizations can empower developers to detect and address security issues early and efficiently, enhancing overall visibility and reducing the risk of vulnerabilities making it into production.

SCM EXTENSIONS: VISIBILITY WITH EVERY CHANGE

Black Duck's out-of-the-box plug-ins for major SCM platforms like GitHub, GitLab, Azure DevOps, Jenkins, and Bitbucket allow developers to optimize test workflows to suit the project at hand while staying aligned with risk tolerance policies. To use these plug-ins, simply adjust the preconfigured parameters included within the templatized SCM workflow automation scripts, which provide simple ways to perform activities such as

- Invoke SAST or SCA scans
- Compare against existing risk tolerance policies
- Automate actions consequent to failed scan or policy violations (e.g., break the build)
- Automate developer feedback mechanisms for issue details and fix guidance (e.g., pull request comment, open fix pull request)

Perhaps the most notable benefits of SCM extensions are the greater visibility they provide into the organization's shared codebase (rather than an individual developer's active working files) and enhanced collaboration during code review, merge, and release phases.

By using both IDE plug-ins and SCM extensions, developers benefit from immediate, local assistance while coding as well as centralized, team-wide control and visibility over the codebase. This leads to better security with less friction.

Together, these mechanisms make security-approved pipelines the path of least resistance and remove the incentive to use unapproved, unmonitored secondary pipelines.

Scenarios that increase visibility

IDE integration

1. Developer writes code and the IDE's SAST scan highlights a potential SQL injection vulnerability.
2. Developer receives a real-time alert and in-depth guidance to fix the issue.
3. Developer declares an open source dependency, then runs a local build.
4. Developer receives a list of declared and transitive dependencies with security vulnerabilities and associated software licenses that may need to be addressed.

SCM extension

1. Developer edits some elements for a commercial, revenue-generating application and commits the code into GitHub.
2. This triggers a security scan that has been configured using the Black Duck security scan GitHub Action.
3. The pipeline scan detects a coding flaw that allows for a cross-site scripting attack and a software license that violates company policies for the commercial application, so it blocks the build.
4. The developer is alerted to the policy violation, reviews the automatically generated pull request comment for the software license conflict, and reviews the automatically generated fix pull request created by Black Duck's AI security assistant for the code flaw.

APPLICATION SECURITY PLATFORMS: VISIBILITY THAT SCALES WITH AI

Modern AppSec tools automate the detection of security issues and the enforcement of policies, helping organizations achieve a high level of security and compliance, streamline the development process, and reduce the burden on developers. These tools also scale to meet the evolving needs of AI-enabled development workflows.

Polaris is a cloud-based AppSec platform optimized for the needs of development and DevSecOps teams. It unites Black Duck's market-leading security analysis engines and offers

- **An integrated application security testing platform.** Polaris provides a single source of truth for SAST, SCA, and dynamic application security testing (DAST) results. When integrated into CI pipelines, Polaris automatically performs analysis and triage based on policies defined by security teams and informs developers what to prioritize for remediation.
- **Centralized policies.** Polaris enables organizations to define and manage security policies all in one place, making it easier to apply them consistently across all stages of the CI/CD pipeline, from initial development to code commit to final deployment. Organizations can also tailor policies to their specific needs, including industry-specific regulations and internal security guidelines.
- **Automated enforcement.** Polaris can perform automated actions based on pipeline triggers, scan completion, or policy violations. These actions can include, for example, automatically blocking pipeline activities if critical security issues are detected, or automatically opening a Jira ticket for issue resolution and assigning it to the responsible developer. These enforcement mechanisms allow Polaris to establish consistent yet flexible security gates that throughout the SDLC.
- **AI-powered AppSec.** Polaris uses a proprietary, security-optimized AI assistant to clarify issue details, help developers fully understand the issue at hand, and generate code fixes for SAST issues.

CREATE COMPREHENSIVE VISIBILITY FOR YOUR TEAM

By adopting these strategies, practices, and tools, you can ensure that issues are detected and prioritized for remediation as close to their inception as possible. And by removing the ambiguity of next steps and closing the feedback loop across teams, you can safeguard the efficiency and resilience of your DevSecOps program as workflows evolve to rely on AI coding assistants.

BLACK DUCK CAN HELP

Black Duck provides a comprehensive portfolio of best-of-breed application security testing solutions, enhanced by deep integrations and dedicated plug-ins that span every stage of the SDLC. Our market-leading testing engines enable organizations to orchestrate, aggregate, standardize, and prioritize findings to gain a complete view of their software risk landscape. And our Code Sight IDE Plug-in and SCM security automation templates and extensions for leading platforms like GitHub, GitLab, and Azure DevOps support the growing number of enterprises leveraging development teams as a first line of defense for software security.

Ready to get started? Contact us today to see how we can help your organization enhance its security visibility to improve risk management.

ABOUT BLACK DUCK

Black Duck® meets the board-level risks of modern software with True Scale Application Security, ensuring uncompromised trust in software for the regulated, AI-powered world. Only Black Duck solutions free organizations from tradeoffs between speed, accuracy, and compliance at scale while eliminating security, regulatory, and licensing risks. Whether in the cloud or on premises, Black Duck is the only choice for securing mission-critical software everywhere code happens. With Black Duck, security leaders can make smarter decisions and unleash business innovation with confidence. Learn more at www.blackduck.com.