BLACKDUCK®

Reduce Friction in Your DevSecOps Program

Overview

Organizations are always looking for ways to go faster. Faster revenue growth, faster global expansion, and faster innovation. Faster development cycles are part and parcel of this journey for speed, but organizations need secure fast development cycles to avoid risking any benefit they may receive from that acceleration. Organizations often face challenges securing their applications without impeding development velocity. Adding distinct point tools to test projects for vulnerabilities is generally the first attempt at application security. While this helps detect potential security threats, it adds many hurdles to the software development life cycle (SDLC). And when developers are in the middle of a sprint, hurdles slow them down.

Too many late-stage tests, tools to manage, and findings to review make it difficult for organizations to gain actionable insight into their risk, and it can introduce a variety of potential points of failure into a DevSecOps program. Handing developers a list of unprioritized issues to fix and last-minute rework greatly impedes their ability to deliver software quickly. Simple, scalable, and flexible application security testing (AST) solutions can help businesses navigate DevOps without compromising on security standards.

To eliminate friction in rapid development cycles, organizations must make security-enabled pipelines the path of least resistance. In this guide, we'll look at three places in your SDLC where friction can most easily be addressed: at a project's inception, during routine commit and build cycles, and in remediation workflows.

Reduce friction when securing new development

The developers' role is to code new functionality that enhances both the software that organizations deliver to customers, and the applications they use in the daily operations of the business. It's imperative that DevSecOps programs balance development's priority on innovation and speed with security's prioritization of code quality and risk reduction.

One opportunity to achieve this is to automatically onboard new projects into security workflows. This ensures that developers' throughput adheres to set standards and risk tolerance thresholds without depending on development teams to declare their work each time they make a new branch. Black Duck® Polaris® Platform allows security and development teams to automatically bulk-onboard entire source code management (SCM) repositories, a capability unmatched by other application security testing tools. This accelerates the process of incorporating new code into testing workflows and ensures comprehensive coverage without additional burden on developers.

Reduce friction from security scans in the pipeline

The second opportunity for organizations to reduce friction is by optimizing both ***how*** and ***where*** application security testing occurs within the SDLC and continuous integration (CI) pipelines. This requires functional integrations between AST solutions, development tools, build environments, and repositories at various stages across the SDLC. You can then leverage security testing policies that define the context for when distinct security tests occur.

For example, you might choose to run static application security testing (SAST) scans only when proprietary code is changed, omitting software composition analysis (SCA) because there were no changes to open source dependencies. You might opt for SCA when new open source libraries are resolved into the build, but omit SAST because there were no changes to proprietary code. You can choose to run incremental analysis of the project files that were modified since the last scan. Optimizing contextual requirements such as these minimizes the delay and resource consumption that comes from running unnecessary tests.

An AST platform allows you to manage policies for each of your security tests from a central location, providing a singular point of integration and minimizing potential points of failure. This helps establish risk tolerance thresholds based on predefined criteria such as pipeline activity, code changes, standardized risk metrics, and compliance standards.

You can eliminate more friction by establishing automated security gates and defining actions to be taken upon policy violation. While the actions may vary depending on where in the pipeline a scan occurred or a violation was noted, an AST platform suited for DevSecOps will allow you to automatically alert developers at commit, open fix pull requests, break builds, or block promotion into binary repositories or production environments.

A frictionless, integrated DevSecOps program can support end-to-end security, fostering risk awareness and security capabilities.

- **In the integrated development environment (IDE):** Empower developers to identify vulnerabilities in proprietary code and open source without leaving the IDE. Placing rapid, incremental analysis of modified files or full projects at developers' fingertips saves time and effort, avoiding unnecessary distraction and the rework that comes from failed late-stage security tests.
- **In SCM repositories:** As code is committed or modified, automated scans are activated, providing complete risk awareness on any assets that may have entered the pipeline without passing earlier security tests. Integrated testing at this stage can also provide early warning of emergent issues such as open source components that had been previously scanned by SCA tools but for which new vulnerabilities have been published. This also helps safeguard against any assets that may have evolved through unapproved secondary development streams that often stem from excessive friction in pipelines managed by well-intentioned but ill-equipped security teams.
- **In build environments:** Performing integrated testing at this stage helps ensure that any unknown issues, which may be resolved into the application during the build, are detected and prioritized for remediation. This helps to minimize risk exposure to any third-party assets that may not adhere to your defined security standards.
- **In preproduction test and quality assurance (QA) environments:** It is possible to gain additional security risk insight without running additional test cycles. Interactive application security testing (IAST) solutions run alongside automated or manual functional and unit testing, allowing developers, QA, and DevOps teams to follow existing workflows while observing the running application and assess insecure or anomalous activity. This helps ensure visibility into issues that manifest only at runtime, which would likely not be evident in earlier static scans.

A centralized AST platform is the most efficient and resilient way to coordinate these tests throughout the pipeline. The Polaris platform, for example, supports concurrent scanning for various tests, such as SAST and SCA, abbreviating test cycles. An AST platform further reduces friction by consolidating, cleansing, and prioritizing security risk insight from these diverse tests, reducing distraction for security and development teams tasked with remediation.

Reduce friction for faster remediation

The third opportunity to reduce friction in your SDLC comes at the remediation stage. A key to efficient remediation is establishing closed feedback loops with developers, reducing the time between risk detection, prioritization, and fix requests. Integration and automation are essential for this. Automated security tests derive valuable information across different pipeline stages, with policies providing clear and standardized prioritization of risk, and connections into remediation workflows delivering risk insights directly into developer workflows.

Examples of optimized remediation mechanisms between security and development teams include

- **Issue management integration:** By integrating with issue management tools (e.g., Jira), prioritized security findings seamlessly flow into development workflows. Developers receive direct notification of new risks, including clear remediation guidance and patch recommendations, such as those provided by the Black Duck Cybersecurity Research Center (CyRC). Bidirectional integration between issue management systems and the AST platform ensures that security teams are kept apprised of the fix status of assigned issues, providing a more timely and accurate assessment of risk exposure for a given project.
- **IDE-based feedback from pipeline scans:** It is possible to provide direct access to prioritized issues within the developer's preferred IDE. This is made possible with the Black Duck Code Sight™ IDE plugin and analysis from the Polaris platform. While an IDE security plugin can enable developers to perform their own rapid, local security scans, a greater breadth and depth of analysis is possible from pipeline-based analysis. This gives developers greater insight into risks that are resolved into the build, such as transitive dependencies, or issues that violate policies defined by security teams and enforced during later stages.
- **Developer security training:** While development activities increase in complexity and frequency, the fact is, the security capabilities among developers are not standardized across individuals or teams. Secure coding education can accelerate remediation by providing short, modular training related to specific risks detected during security tests. This may also alleviate the burden on security teams by precluding new issues with higher-quality code, ultimately avoiding rework consequent to failed tests. Developer security training, such as that which Black Duck provides with its partner Secure Code Warrior, can be prescribed as a tailored curriculum or as-needed, accessible within the IDE or issue management integrations.

How Black Duck can help

DevSecOps is accelerating development for the next generation of software and cloud-native assets, and security is tasked with keeping up without sacrificing coverage or impeding DevOps pipelines. Complex development workflows, diverse testing tools, and distributed teams continue to be a challenge, but integrating testing technologies, defining contextual policies, and automating remediation workflows are emerging as the best mechanisms for balancing efficacy and efficiency.

The success of your DevSecOps program relies on centralizing high-quality risk data, ensuring complete testing coverage that's appropriate for the software passing through the pipeline, and establishing a strategy that can scale as your organization grows.

The Black Duck portfolio of application security testing solutions, developer-focused IDE integrations, and scalable software integrity platform establishes an end-to-end strategy for integrating security into DevOps workflows and CI/CD pipelines.

Learn more about developer-first security

View webinar series

About Black Duck

Black Duck® offers the most comprehensive, powerful, and trusted portfolio of application security solutions in the industry. We have an unmatched track record of helping organizations around the world secure their software quickly, integrate security efficiently in their development environments, and safely innovate with new technologies. As the recognized leaders, experts, and innovators in software security, Black Duck has everything you need to build trust in your software. Learn more at www.blackduck.com.

©2024 Black Duck Software, Inc. All rights reserved. Black Duck is a trademark of Black Duck Software, Inc. in the United States and other countries. All other names mentioned herein are trademarks or registered trademarks of their respective owners. September 2024