Expand Your Risk Awareness to Accelerate Application Security

Overview

Lack of visibility into security issues is one of the biggest problems that DevSecOps programs face as they look to increase application security (AppSec). The only way to address it is a concerted effort by development, DevOps, and AppSec teams to gain a comprehensive understanding of risk across the board. However, a few key challenges continue to impede DevSecOps initiatives.

The first challenge is the sheer diversity and velocity of development activities. Each project is composed of open source libraries and proprietary code, with third-party assets frequently in the mix. These projects are built to run on-prem or in the cloud, in monolithic, containerized, or serverless architectures, with each developer often focusing on a different set of technologies and functions, further isolating visibility into any given project's security posture.

The second challenge is the complexity of continuous integration (CI) and continuous deployment (CD) pipelines, as the spectrum of tools and technologies used in modern applications makes it difficult to establish consistent mechanisms to detect and prioritize issues.

Finally, the absence of a culture of DevSecOps can obscure visibility into hidden problems as security is deprioritized in exchange for speed or because of reduced resources. The intricate nature of software requires collaborative problem-solving, and individual developers may lack the insight or resources to tackle security-related issues independently.

Ultimately, heightened risk awareness and improved visibility across the software development life cycle (SDLC) is the key to addressing security concerns and ensuring the integrity of software systems.

Coming up with a strategy

If you can't find them, you can't fix them, so how do you begin to address hidden security issues? A few key strategies can help organizations expand their risk awareness and strengthen their security posture in a way that suits both CI pipelines and DevOps workflows.

- Put the mechanisms in place to establish end-to-end visibility, giving contributors across different teams a clear line of sight into risks as they code and as third-party components are resolved into their builds.
- Eliminate subjective assessment of "riskiness," "priority," and "security" by standardizing developers' security capabilities and terms.
- Close feedback loops and disseminate risk insight by using functional integrations and feeding prioritized information to developers through established issue management workflows.

Establish end-to-end visibility

To instill effective security practices in your DevSecOps program, begin by establishing end-to-end visibility across all stages of the SDLC. For every team, a principal goal at this stage should be to gain a clear line of sight into issues as early as possible, but there can be secondary goals as well.

- For development teams, early visibility into issues is critical, ideally as they code, declare dependencies, and check artifacts into source code management (SCM) systems and repositories. Visibility into fix priority will also help define a clear path to resolve any issues flagged during local or pipeline-based testing that may otherwise prevent the artifact from being promoted downstream or into production.
- For DevOps teams, consistent visibility across the SDLC is top priority, allowing them both to attest to the current security status of the artifacts passing through pipelines and to make informed, incremental adjustments that will help assure timely and secure software releases. They should also have visibility into functional and unit tests, as well as security scans.
- For AppSec teams, comprehensive visibility into development projects, tools, and assets ingested through the software supply chain helps inform decisions around application security testing (AST). AppSec teams benefit from additional insight into regulatory requirements, secure coding best practices, and the expectations of customers, partners, and stakeholders.

Cross-team discussions help guide security teams, who often still own DevSecOps initiatives and who are supported by development and DevOps teams with ongoing implementation. The output of such collaboration should include a tactical approach to DevSecOps, which is where tools can help.

- **Black Duck® Polaris® Platform.** Polaris provides developers with visibility into risks detected during pipeline scans and extends visibility into remediation recommendations based on prioritized risks. By using this as-a-service, cloud-based AST solution, you can establish a single source of truth for detected security issues in proprietary code, open source, and third-party artifacts.
- **Black Duck integrations and developer plugins.** Out-of-the-box resources for popular platforms like GitHub, GitLab, and Azure DevOps are part of a suite of AST solutions that enable security teams to dictate where and how security tests run automatically. Throughout the pipeline, you can balance test coverage with velocity while maintaining full visibility into risk at every stage.
- **Black Duck Code Sight™ IDE plugin**. This plugin provides on-demand or automated static application security testing (SAST) and software composition analysis (SCA), along with issue details and remediation guidance, to establish an effective "security spell-checker" for developers. Empower developers to detect issues in proprietary code and vulnerable open source components without leaving their preferred integrated development environment (IDE).
- **Black Duck Seeker® Interactive Analysis.** Want to gain security-related insight from existing preproduction tests run by development and QA teams? This tool also helps AppSec teams expand visibility to include potential risks to sensitive information, with an additional benefit to legal and compliance teams who must align software to industry or regulatory standards.

Eliminate subjective assessments

Aligning definitions of "risk" and "security" to standards or guidelines is often the first step for organizations structuring a modern DevSecOps initiative. These definitions should be clarified across teams and tools, with automatic enforcement of corrective measures to stifle propagation of misaligned terms.

Even with clear and consistent definitions, though, developers often vary wildly in their secure coding capabilities and their ability to effectively and efficiently detect security issues inadvertently introduced into project branches. By integrating security training delivered in modular chunks related to the risk detected in the development pipeline, developers can improve their skills while addressing real issues. This offers the compounded benefits of accelerating remediation, abbreviating windows of opportunity for attack, and precluding future software security issues.

Defining risk tolerance thresholds, prioritizing issues based on organizational standards, and implementing flexible policies across your portfolio can also help ensure that your risk tolerance and security policies are consistently applied across the organization.

To achieve this, you should

- **Centralize flexible policies.** The Polaris platform can centralize policies governing multiple types of testing across various pipeline integration points, ensuring consistency and adherence to organizational standards while alleviating the management burden of AppSec tooling.
- **Automatically enforce risk tolerance thresholds.** Policies can establish which factors will block pipeline activities (e.g., SCM check-in, build, promotion into staging/production) and which are acceptable.

- **Validate true risks that manifest at runtime**. Tools such as Seeker IAST and its automatic risk validation capabilities can retest issues detected during preproduction tests.
- **Provide modular, risk-relevant secure coding training.** Educating developers on secure coding practices to improve their skills can reduce the risk of introducing new vulnerabilities into the codebase. This can be done in an agile, responsive manner by assigning fix guidance to specific developers tasked with addressing issues detected and prioritized during pipeline scans.

Close feedback loops

Once visibility is established and the team is aligned to standardized assessment of risk and security, disseminating the insight necessary to quickly fix issues becomes crucial. Simply put, developers need assistance in addressing these issues, and delaying such guidance extends the window of opportunity for an attack and threatens DevOps teams with missed shipping deadlines.

Maintaining a closed feedback loop among contributors allows teams to manage security risks at the scale and velocity required by both DevOps and CI pipelines, with any fixes undergoing retesting and validation against established policies to ensure compliance. The following is an example of what an effective feedback loop might look like:

- **Alert developers to new issues.** Automatically perform appropriate security testing across the SDLC and CI pipeline using the Polaris platform. Minimize delays and unnecessary distraction by running scans based on pipeline activities or code changes. Integrate with issue management tools like Slack and Jira for seamless communication. Enable developers to perform complementary testing local to their IDEs with the Code Sight plugin.
- **Provide comprehensive fix guidance.** Ensure developers have access to recommendations for code changes, patches, mitigating factors, and workarounds. Give them access to up-to-date fix intelligence written and curated by the Black Duck Cybersecurity Research Center (CyRC) or educational materials from Black Duck Developer Security Training, powered by Secure Code Warrior.
- **Leverage bidirectional issue management integration.** Close security feedback loops with the Polaris platform, ensuring a continuous understanding of the project's security posture and progress in addressing identified issues. Automate new issue alerts to developers and fix attestations to security teams through bidirectional issue management integrations so that AppSec dashboards always reflect the most current security posture.

Creating comprehensive visibility for your team

By adopting these strategies, you can ensure that issues are detected and prioritized for remediation as close to their inception into the codebase as possible. By removing the ambiguity of next steps and closing the feedback loop across teams, you can ensure the resilience of your DevSecOps program as workflows and projects evolve.

The first step of identifying security risks should flow across all projects and pipelines. Doing this efficiently means testing at different stages throughout your SDLC for end-to-end visibility and to establish safeguards that account for unapproved or unknown secondary pipelines as well as artifacts ingested through third-party supply chains.

The next step is making remediation standards transparent to developers and DevOps through policies and automated security gates. The final step is to close feedback loops as quickly as possible by leveraging deep integration with issue management workflows.

How Black Duck can help

Black Duck provides a comprehensive portfolio of best-of-breed AST solutions, enhanced by deep integrations and dedicated plugins that span every stage of the SDLC. With premier testing engines and policies centralized atop the Black Duck Polaris Platform, organizations can aggregate, standardize, and prioritize findings to establish a complete view of their software risk landscape. Black Duck supports the growing number of enterprises leveraging development teams as a first line of defense for software security risks with the Code Sight IDE plugin and SCM security automation templates and extensions for leading platforms like GitHub, GitLab, and Azure DevOps.

Learn more about how Black Duck can help contributors from development, DevOps, QA, and security teams simplify applications security testing without sacrificing coverage or speed.

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.