



FASTER AS THE CATALYST FOR DEVSECOPS TRANSFORMATION

MINIMIZE THE ATTACK SURFACE WITHOUT
SLOWING DOWN DEVELOPMENT



Organizations are under relentless pressure to provide or expand digital services to meet customer needs and business goals. Keeping up means working quickly—and putting out applications at speed. Methodologies such as agile development and DevOps have emerged to meet this demand and stay competitive with digital-first disrupters such as Amazon, Airbnb, Netflix, and Uber. Indeed, digital-first companies are 64% more likely to exceed their business goals compared to their peers.¹

But with success comes new challenges—especially in the area of security. Organizations on the bleeding edge of DevOps need to build security into their processes, workflows, and tool integrations throughout the software development life cycle (SDLC). But how does an enterprise remove the cross-functional silos and incorporate security seamlessly across the SDLC?

Many organizations are discovering interactive application security testing (IAST) as the catalyst to turn DevOps into DevSecOps.

RECIPE FOR BUSINESS SUCCESS: AUTOMATION, COLLABORATION, SPEED

Enterprises are only as fast to market as their slowest process allows. With agile development and continuous integration and continuous delivery (CI/CD) pipeline models, organizations can develop apps quicker than ever. In this environment of accelerated innovation, DevOps is mandatory. It enables a digital-first model in three important ways: automation, collaboration, and speed.

- **Automation.** Automating as many manual and repetitive tasks as possible reduces development time and the risk of human error, increasing the consistency and quality of the final product.
- **Collaboration.** Breaking down silos between development and operations spreads the responsibility of end-to-end development, and business practices such as organizational hierarchy, isolated departments, and annual strategic planning are less likely to impede processes.
- **Speed.** Increasing speed and agility enables a quick response to market conditions and customer requirements, helping to achieve and maintain competitive advantage.

NEW SUCCESS BRINGS NEW CHALLENGES

An inevitable challenge of building more applications faster is security. Applications are increasingly complex—built and deployed with a mix of containers, microservice models, and open source toolsets, and integrated with third-party components and codebases. Each technology provides a potential path for bad actors to exploit. Every new application expands the attack surface and must be built with that in mind.

“Web and microservices-based apps are the top two apps increasingly used and developed. With these apps here to stay for the foreseeable future, the number of unanticipated exploits will continue to grow.”

—Kimm Yeo, Black Duck



Some of the advantages of DevOps—development speed, use of cloud platforms, and collaboration—also introduce risk.

But some of the advantages of DevOps—development speed, use of cloud platforms, and collaboration—also introduce risk.

Development speed

Reducing the time from the discovery of a risk or vulnerability to remediation is key to shortening development time. But this shorter feedback loop can lead to missed errors and vulnerabilities. Traditional security tasks, such as analyzing code, checking configurations, and assessing for known vulnerabilities, take time. Too often, development speed outpaces what the security team can do.

Coding mistakes are a natural outcome of increased speed and complexity. Paradoxically, these mistakes lead to slower development cycles. The more code there is, the more chance of errors, and the more time it takes to identify and fix the vulnerabilities, which can delay deployment.

Cloud platforms

Cloud deployments can amplify security issues and slow down operations, especially when the platform is part of the application development process. Unfortunately, even with due diligence into security policies and standards when choosing a cloud provider, users don't have end-to-end visibility into this shared infrastructure and can't be sure how the platform's security works until deployment.

Migration to cloud platforms also exposes data to the potential of being lost, leaked, visible to third parties, and otherwise compromised.

Collaboration

DevOps requires constant collaboration between the development and operations teams, each of which is accustomed to working within its own processes, policies, and standards. Failure to explicitly define team roles and responsibilities can result in security gaps.

Collaborating effectively requires the sharing of data, including sensitive and privileged information such as account credentials, tokens, and SSH keys. This data is shared between teams and their systems, applications, containers, and microservices. Failure to properly manage this confidential information is another path for attackers to steal information or disrupt operations.

THE SOLUTION: DEVSECOPS

Just as DevOps broke down the silo between development and operations and expedited application production and speed to market, DevSecOps removes security as a silo by shifting it left into the design and testing phases, and continues it through to release. This means speed and complexity don't result in more vulnerabilities, lower quality, and greater risk.

More and more organizations are adopting this approach. According to Gartner research, by 2022, 90% of all software development projects will be following DevSecOps practices (up from 40% in 2019).²

IAST as catalyst for DevSecOps transformation

IAST is one of the few AppSec tools that can support a digital-first strategy without adding time to the process. In other words, it minimizes the attack surface without slowing down development.

Enterprises use hundreds of apps and update or release new ones on a weekly or daily basis. Developers depend on automated and integrated code delivery to keep pace. This rapid development velocity adds to the complexity of modern apps and increases security challenges. Web applications are the most common vector of security breaches such as SQL injection, cross-site scripting, and sensitive data leakage.³

“It is important to reassert that this trend of having web applications as the vector of attacks is not going away. This is associated with the shift of valuable data to the cloud, including email accounts and business-related processes.”

—Verizon 2020 Data Breach Investigations Report

To ensure the security of web applications, development and security teams need to do more than static application security testing (SAST) and software composition analysis (SCA). These types of tests identify security weaknesses and vulnerabilities in proprietary and open source code respectively, but only in static environments. They can't anticipate critical vulnerabilities triggered during runtime compilation and execution.

Dynamic application security testing (DAST), like IAST, provides dynamic testing during various stages. Unlike IAST, however, DAST is a black-box test and requires time and subject matter expertise to triage findings and troubleshoot remediation. This doesn't work for CI/CD pipelines in DevOps environments.

DAST

Pros

- Tests at runtime
- Shows how app will actually respond to vulnerabilities
- Doesn't require source code

Cons

- Tests from outside, has no knowledge of internal workings
- Simulates attacks, doesn't test real-world conditions
- Has no visibility into program
- Doesn't integrate well in CI/CD

SAST

Pros

- Analyzes applications from the inside out, early in the SDLC
- Identifies exact lines of code to address
- Integrates in CI/CD

Cons

- Doesn't check runtime behavior

PEN TESTING

Pros

- Identifies exact lines of code to address
- Finds vulnerabilities that can't be found via automated tests

Cons

- Simulates attacks, doesn't test real-world conditions
- Tests are labor-intensive and time-consuming

WHY IAST

Many vulnerabilities can be detected only by testing code dynamically in the running application. IAST bridges the gap between existing testing techniques like SAST, SCA, DAST, and pen testing. It provides real-time results in seconds, so it's the only type of AppSec testing that can keep up with DevSecOps and quick-turnaround CI/CD processes.

IAST:

- Enables a shift left of dynamic application security testing to catch runtime vulnerabilities earlier
- Provides developers with stack traces, source location, lines of code, and contextual remediation advice
- Allows identification and remediation of vulnerabilities before production

With its focus on runtime behavior, IAST doesn't need source code to detect vulnerabilities. It identifies specific lines of compromised code, so developers can focus on fixing high-priority issues.

Vulnerabilities that can only be detected at runtime with IAST

- 
- Sensitive data leakage in value or name patterns
 - Information disclosure via HTTP headers
 - Weak SSL encryption
 - Server misconfigurations
 - Cross-microservices dataflow mapping and tainted data
 - Unused/hidden parameters that can be easily exploited

IAST BENEFITS: A WIN-WIN FOR DEVELOPERS AND SECURITY

Using IAST to enable DevSecOps benefits the entire enterprise by providing better security at lower cost—without sacrificing speed to market. IAST decreases cost by empowering developers to fix vulnerabilities early in the development process, when it's significantly less costly to address them. This also means there's no need to wait for additional scans, verifications, and validation of vulnerabilities, expediting overall development time. And by providing developers with clear remediation guidance, IAST removes the need for a separate AppSec team as part of the critical path. It's no surprise that, compared to penetration testing, IAST reduces remediation time by 65%.⁴

Developer benefits

IAST allows development teams to test and troubleshoot issues quickly, and to seamlessly integrate security into the SDLC—all of which are essential to a DevSecOps methodology. IAST offers four key benefits to developers: speed, accuracy, a holistic approach, and seamless integration.

Speed

Because IAST continuously monitors applications and detects vulnerabilities in real time in the background during functional testing, it doesn't delay the development and release process. It enables developers to remediate issues quickly, providing feedback on where to find the vulnerability in source code or component libraries, and detailed advice on how to fix it. Additionally, by receiving feedback early in the cycle, when developers are most familiar with their code, it's easier and less costly to implement fixes.

Accuracy

Developers waste valuable time chasing false positives. IAST provides accurate, risk-prioritized test results and lower false positive rates, requiring fewer staff hours to review and remediate.

Holistic approach

Because there's no additional work required outside the development process, IAST is useful across all phases of the SDLC—from developing code to quality assurance (QA), functional testing, and moving the product into production. The reliability and specificity of IAST findings allow QA testers to quickly identify security vulnerabilities without extensive AppSec experience. And the ability to fix risks in real time means a negligible impact on business performance.

Seamless integration

IAST seamlessly integrates into existing CI/CD pipelines, making it easy to deploy, update, and scale to support enterprise needs.

AppSec team benefits

IAST frees the AppSec team from having to deal with easily fixed vulnerabilities and errors. They can focus DAST and pen testing resources on more difficult, corner-case vulnerabilities that require more intensive, manual testing to identify and verify. This allows them to focus on strategic initiatives rather than being reactive.

IAST also produces a comprehensive overview of the risk posture for the security team, including:

- Compliance reports for standards such as OWASP Top 10, PCI DSS, CWE/SANS Top 25, and GDPR
- Vulnerability reports aggregated by severity
- Common attack pattern enumeration and classification (CAPEC) taxonomy

USE CASES

IAST enables DevSecOps teams to build in continuous runtime security as part of the CI/CD. This provides benefits to development teams, QA teams, and production teams.

Development

By finding and addressing security issues early in the SDLC, IAST lowers remediation costs and reduces reliance on AppSec resources and expertise to triage, verify, and troubleshoot—especially with known vulnerabilities or highly repetitive tasks. Organizations should look for tools that work with the build integration and task management systems the team is already using, such as Jira and Jenkins, as well as those that integrate with IDEs to provide immediate feedback.

Quality assurance

With IAST, QA teams don't need to introduce additional security checkpoints. AppSec testing is done dynamically and concurrently within the normal scope of tests carried out by the QA team. Because it doesn't introduce more security scans or security gates or checkpoints, IAST is ideal for fully automated functional tests, and continuous testing in a CI/CD environment.

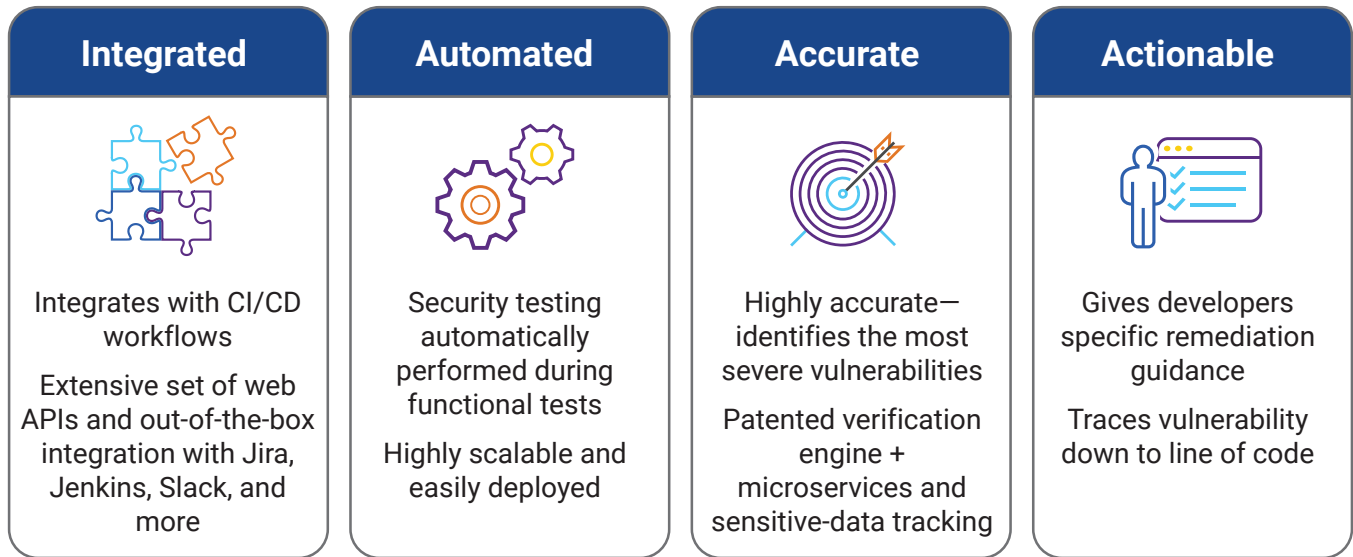
Production

It's nearly impossible to fix all vulnerabilities before going to production, so when an application reaches this stage, it's valuable to know which issues are the most serious. IAST helps prioritize the detected vulnerabilities according to risk severities defined by the organization. It can help monitor, detect, verify, and alert on critical vulnerabilities and risks.

IAST seamlessly integrates into existing CI/CD pipelines, making it easy to deploy, update, and scale to support enterprise needs.

SEEKER: THE INDUSTRY-LEADING IAST SOLUTION

Seeker® Interactive Analysis is an award-winning IAST tool from Black Duck that helps development, QA, DevOps, and security teams automate the security testing of modern applications (web-based, cloud-based, microservices-based, etc.). It's the industry's first IAST solution with patented active verification and sensitive-data tracking capabilities. It's accurate, easy to use, and scales to support enterprise needs while identifying and verifying vulnerabilities in real time. And where other IAST solutions stop at detecting and reporting, Seeker goes a step further by automatically verifying and prioritizing findings. It instantly reports the vulnerabilities that matter to your organization.



References

- ¹ Adobe in association with Econsultancy, [Digital Trends report](#), 2019.
- ² Gartner, [Integrating Security Into the DevSecOps Toolchain](#), 2019.
- ³ Verizon, [Data Breach Investigations Report](#), 2020.
- ⁴ Forrester, [Business Technographics Global Security Survey](#), 2020.

See how Seeker IAST can help your organization maintain development velocity and application quality while maximizing security.

[Request a demo](#)

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.