SYNOPSYS®

AppSec Strategy Guide: A Step-by-Step Breakdown to Achieve DevSecOps

Executive Summary

As development technologies become more fast-paced, modular, and automated, the tools and practices used to secure the software must also evolve. While many application security testing (AST) tools can be integrated into the pipelines, teams often struggle with complexity, performance, and noisy results. Injecting security into DevOps without sacrificing efficiency requires a concerted approach focusing on

- Equipping developers with what they need to secure code as they write it
- Ensuring that integration and automation minimizes impediments and runs the necessary tests at appropriate times
- Automating modular AST services that perform continuous testing and verification in the CI/CD pipeline
- Remediating prioritized risks aligned to business needs

Organizations need to adopt a new approach that provides intelligent, context-aware application security risk management.

The Status Quo: Velocity over Security

As organizations continue to adopt DevOps, the speed and complexity of software development has accelerated. In response, security teams are working in tandem to streamline testing by integrating application security testing (AST) tools into DevOps workflows. But integrating security inefficiently can introduce hurdles that offset the time savings of DevOps, such as wading through numerous or redundant findings, extraneous testing, or an inability to triage or understand how to remediate known vulnerabilities. These challenges have caused many DevOps initiatives to stall or fail, leaving applications less than fully tested—and less than fully secure.

For this reason, applications have become an attractive vector for cybercriminals to target, and that makes software risk a business risk that extends all the way to a company's bottom line. Many web, mobile, and microservice applications reside beyond the firewall, but they still provide access to sensitive data and other systems inside the protected network. Cybercriminals have learned that it's often easier to target vulnerable applications than an organization's network infrastructure. Indeed, nearly 50% of all data breaches over the last several years have exploited application vulnerabilities.[1]

These breaches are also increasingly expensive. The average financial impact of a data breach rose from $3.86 million in 2020 to $4.82 million in 2022—the highest average in the 18-year history of the IBM "Cost of a Data Breach Report."[2]

These statistics make the findings of a recent study by Enterprise Strategy Group (ESG) all the more alarming: 79% of organizations admitted to pushing application changes into production with *known vulnerabilities*. When asked why, 54% said the need to meet critical deadlines forced teams to prioritize releasing vulnerable code instead of fixing it. This, despite 70% of the same organizations reporting that they use 11 or more AST tools at any given time.[3]

Clearly, the legacy approach to application security (AppSec), which "tacks on" testing, triage, and remediation to DevOps, is not keeping pace with modern software development or the threats presented by today's cybercriminals.

A new approach to AppSec is needed—one that addresses business risk without impeding business progress, removes the false choice between speed and security, and makes the promise of DevSecOps a reality.

The New Generation of AppSec: Achieving Security Velocity

To build security into DevOps and achieve true DevSecOps, organizations need to manage AppSec workflows without hindering speed and flexibility. This requires integrating AppSec at every stage of the software development life cycle (SDLC) and giving security and development teams a global view of software risk, critical vulnerabilities, and workflow management across tools, roles, and operations. Doing so can enable intelligent, context-aware application security risk management.

But how do you get there?

Building security into DevOps means defining policies-as-code that define how security testing should work, and implementing them so that security occurs automatically as your other automated DevOps pipeline scripts perform their tasks.

There are three mechanisms that will help your organization achieve this.

- Policies are the first mechanism you'll want to rely on when minimizing friction in the DevOps workflow while still maximizing the impact on your risk posture. Setting good policies will help ensure that the right tests run at the right time without slowing down your DevOps pipelines.
- The next mechanism that elevates your risk management capabilities is one that orchestrates consolidates security findings from automated and manual tests into single place, removes the noise, and calls out the smaller set of high-priority issues. This mechanism also deduplicates test results, removes false positives, and eliminates risks that don't manifest in the context of the application or environment.
- The third mechanism for fostering secure development is one that feeds these clean, prioritized testing results directly to existing developer workflows by delivering it right into the IDE. By putting information directly in the hands of the people who are tasked with fixing the issues, security moves left, and your code becomes more secure.

Ultimately, AppSec velocity relies on using automation to take pressure off three areas: policies and testing, the triage stage, and the coding stage.

Empower Your Developers to Secure Code as Fast as They Write It

The lowest-cost vulnerability to remediate is the one that never makes it into the codebase. Giving developers tools that allow them to fix issues before they commit their code to the build pipeline reduces strain on downstream testing and precludes potential issues in production.

But most developers are not security experts. And tools that are optimized for security teams are often too complex and disruptive to be embraced by developers. To make matters worse, security tools often require developers to leave their integrated development environments (IDEs) to analyze issues and determine appropriate fixes. Few things are more damaging to productivity than constantly switching tools and contexts.

The solution: Fast, lightweight AppSec analysis in the IDE

Developers need an IDE-based AppSec solution that helps them find and fix security issues on the fly, while they code, without switching tools or interrupting their workflows. The solution should combine integrated static application security testing (SAST) and integrated software composition analysis (SCA) to provide real-time alerts and visibility into security weaknesses in proprietary code and known vulnerabilities in open source dependencies.

Code Sight by Synopsys

Code Sight™ offers these benefits alongside insight into unsecured infrastructure-as-code (IaC) configurations, potential secrets or sensitive data leakage risks, and vulnerable API usage.

Code Sight analyzes large codebases in seconds; it scans WebGoat in 3 seconds and Apache Hadoop in 10 seconds. It offers detailed remediation guidance directly in the IDE, helping developers fix issues quickly and improve code quality, even when they lack adequate security training to do so unassisted. Code Sight makes it possible to preclude potential security risks from manifesting in production by ensuring that developers address risks before pushing them downstream.

Code Sight puts security risk insight directly in the hands of the developer, with an integrated SAST engine that can automatically scan and analyze source code and IaC files as developers work. Or it can be manually triggered for on-demand code review. It highlights detected issues in the editor window for easy identification and provides detailed descriptions and remediation guidance, allowing developers to fix many vulnerabilities with a single click.

Code Sight's integrated SCA engine detects known security vulnerabilities in both direct and transitive open source dependencies, identifying them by the Common Vulnerabilities and Exposures (CVE) number or Black Duck® Security Advisory ID. Vulnerabilities are categorized with severity information based on Common Vulnerability Scoring System (CVSS) scores to help prioritize which issues to fix first, without deviating from established risk tolerances dictated by security teams. Remediation guidance helps developers select the next available vulnerability-free or lower-risk version of the component to support security while maintaining the developer's functional requirements.

Code Sight is unique in that it embeds market-leading open source and code analysis technology, optimized for the speed requirements of developers, directly within the tools they are already using. It proactively improves an organization's security posture while saving time and money.

Get a free trial of Code Sight by Synopsys

Try now

Run the Right Test at the Right Time

Choosing the right type of AST tool to use requires several considerations: the environment where the tool is deployed, the types of software flaws it searches for, the programming languages the tool is compatible with, and the stage of the SDLC when testing is run. For this reason, most organizations use a variety of AST tools, including dynamic application security testing (DAST), interactive application security testing (IAST), SAST, SCA, and third-party/commercial tools. According to ESG, 70% of firms use more than 11 AST tools, and 27% use more than 25 AST tools.[4]

Although many tools offer direct integration with DevOps pipelines, teams often struggle with their complex up-front configurations and resulting changes to established workflows. Automating full scans with every build can clog pipelines and overwhelm developers with "noisy" test results without clear priority or relevance to the task at hand.

The challenges of integrating and automating AST tools include

- **Lengthy scan cycles.** DevOps build pipelines run in a matter of seconds to a few minutes, but AppSec tool scans often take several hours. Factor in multiple forms of analysis (SAST, DAST, SCA, etc.), and the problem is compounded, turning remediation hours into days or even weeks.
- **Too many findings.** Integrating and automating full AST scans into continuous integration pipelines causes an overwhelming volume of results, even if only small percentages are problematic enough to require developer attention. Teams get bogged down in triage and remediation, leading to delivery schedules taking precedence over security concerns.
- **Proliferation of tools and scans.** Running multiple testing tools at different points of the SDLC can delay artifacts' procession through the pipeline and produce redundant results that need to be correlated and deduplicated later. Most teams fail to merge related findings, increasing the backlog of remediation activities.

Smaller, purpose-built tests that can be run intelligently–at the right time, to the right depth, and on the right application–relieve congestion and keep DevOps pipelines running smoothly.

The solution: Application security testing orchestration

Smaller, purpose-built tests that can be run intelligently—at the right time, to the right depth, and on the right application—relieve congestion and keep DevOps pipelines running smoothly. To do this, organizations need a mechanism to orchestrate AST solutions, allowing disparate tools and processes to function in a coordinated manner with automated execution.

AST orchestration solutions integrate security tooling across the SDLC by acting as middleware between

- **Development,** including IDEs, CI/CD systems, and bug-tracking
- **Operations,** including container orchestration engines and continuous configuration automation
- **Security,** including scanning tools and vulnerability management

Per Gartner, AST orchestration solutions "aid security, development, and operations teams in coordinating the many security tests that should be performed on code. As such, these solutions can be a significant enabler in implementing DevSecOps initiatives, and they promise substantial benefits in terms of more consistent testing and smoother operations."[5]

In practice, these orchestration solutions automatically run the right security tools or trigger manual testing activities based on defined criteria, such as significant code changes, total risk score, and your organization's security policies.

Intelligent Orchestration by Synopsys

With Intelligent Orchestration, you gain the option of running your AST orchestration solution directly in the build/release pipeline; in a separate, isolated pipeline; or through a separate execution environment. The isolated pipeline runs parallel to existing pipelines and integrates into them via APIs (see Figure 1 below). You also get the added benefit of integrating with third-party tools, whether on-premises or in the cloud.

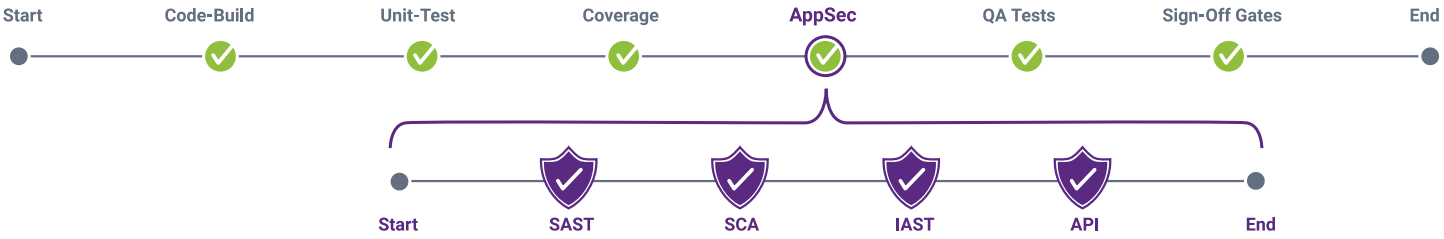

Figure 1. The isolated testing pipeline

Intelligent Orchestration also allows organizations to set policies-as-code, defining the rules for which tests to run and when, and enact those policies programmatically via API. This allows teams to extract security risk insight and establish automated testing and response without encumbering the pipeline with unnecessary processes. For example, you can set risk scores based on criteria such as whether an application is internet-facing, business-critical, contains restricted data, includes critical open vulnerabilities, or has had significant code changes. You can customize the score ranges and types of tools to run based on predetermined policies, compliance, and governance requirements.

Intelligent Orchestration can also initiate manual or out-of-band activities, such as code reviews and penetration tests, through existing defect-tracking systems and communication channels. This enables security and development teams to implement coordinated workflows that align security compliance objectives with application development and release milestones. Figure 2 shows what it looks like to run the right tests at the right time through the SDLC.

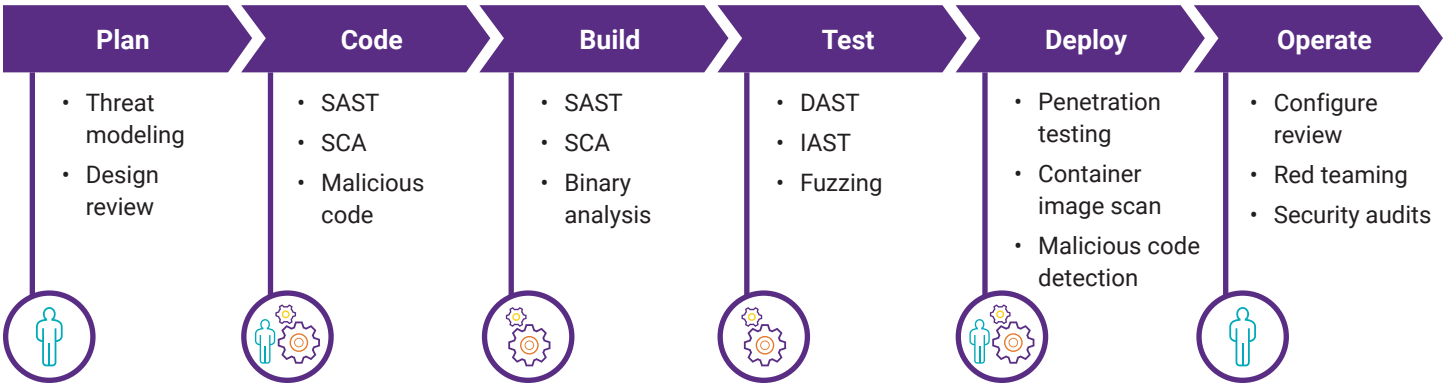

Figure 2. Run the right test at the right time, as well as initiate manual and out-of-band activities

Ultimately, AST orchestration solutions help establish appropriate security control points throughout DevOps pipelines and across the SDLC, providing value to contributors along the way. This includes

- **Developers** who spend less time chasing low-priority defects and more time fixing the ones that present the highest business risk
- **DevOps engineers** who add security checks into their existing workflows without breaking them or slowing them down
- **AppSec teams** who ensure compliance with risk policies and maintain visibility into, and control over, security risks without adversely affecting development and DevOps activities

Automate Security Testing of Modern Web, Microservices, and APIs

Modern software development is based on a complex, distributed computing model that includes microservices, serverless functions, and cloud-native systems. This can make it difficult to identify all the endpoints involved in your system, or trace all the API calls. Additionally, the absence of common standards for APIs compounds this struggle. While the distributed computing framework provides speed and agility for your development team, the proliferation of APIs means that organizations have a much wider attack surface that third parties could potentially exploit.

Organizations need solutions that provide a visual map of the data flow for both inbound and outbound API calls and service endpoints, that can quickly pinpoint potential vulnerabilities, and that have mechanisms to automatically verify results so that real, high-severity vulnerabilities can be prioritized for immediate remediation. Unfortunately, traditional scanners cannot catch, verify, and report critical vulnerability in real time and add friction to the CI/CD pipelines.

The solution: Interactive application security testing

Interactive application security testing (IAST) solutions help organizations identify and manage the security risks associated with vulnerabilities that are discovered in running web applications using dynamic testing (often referred to as runtime testing) techniques. IAST tools monitor an application as it runs, gathering information about what it does and how it performs. IAST solutions continuously monitor and test all application interactions initiated by whatever combination manual and automated tests your organization uses to identify vulnerabilities in real time. IAST does not add to the test workload or cycles as it runs its security testing in the background.

This approach allows IAST solutions to function automatically as part of established functional and security testing workflows, performing autoverification on detected vulnerabilities to minimize noise and highlight true risks. This flexible deployment as part of the existing test workload enables an organization with IAST baked into its DevOps pipelines to effectively turn any functional tests into security tests without adding steps.

The advantages of IAST solutions include

- Sensitive dataflow mapping, API and endpoint discovery, and test coverage results that ensure context-aware analysis to account not only for the type of data and technologies used by the application, but how such resources act and interact
- Automatic app security tests that run in the background with no additional expert resources or scans required, and that run at the same time as your other testing and CI/CD workflows
- Real-time insight into an application's runtime behavior, open source components, and other code-level details
- The ability to qualify the completeness and effectiveness of tests against established security standards, such as the OWASP Top 10 2021, and PCI DSS, GDPR, CAPEC, and others
- Integration with your ticketing system to send confirmation and feedback to your teams

Seeker by Synopsys

Seeker® by Synopsys is an IAST solution that provides unparalleled visibility into your web application security posture, and it identifies vulnerability trends against compliance standards like OWASP Top 10, PCI DSS, GDPR, CAPEC, and CWE/SANS Top 25. Seeker enables security teams to identify and track sensitive data to ensure that it is handled securely and not stored in log files or databases with weak or no encryption. And Seeker's seamless integration into CI/CD pipelines enables continuous application security testing and verification without impeding DevOps workflows.

Seeker is purpose-built for secure DevOps. It uses instrumentation techniques and runtime analysis to continuously monitor, identify, and verify security vulnerabilities in web applications, typically during integration testing and QA, or in preproduction deployment. Whether applications are run on-premises or in cloud environments, are composed of microservices and serverless functions, or make extensive use of APIs, Seeker supports all modern application development needs. Seeker's agent can track every test action performed by the running app. Findings are automatically verified, with results presented in real time and without the need for any additional scans.

Active verification increases the effectiveness of risk triage, retesting identified vulnerabilities and validating whether they can be exploited. This helps eliminate false positives and enables security and development teams to prioritize true risks for remediation. Comprehensive analysis provides all the information necessary to address vulnerabilities, including

- A clear explanation of the risk
- Runtime memory values and context
- A technical description
- The vulnerable lines of code
- Relevant, context-based remediation instructions

Detail panes provide clear visibility into the dataflow and the impact of malicious parameters (e.g., dynamic SQL concatenation), as well as at-a-glance identification of whether a vulnerability was determined to be exploitable or a false positive. This saves security teams and penetration testers time, so they can focus their time on those issues that tools can't solve.

Cut Through Noisy Findings and Focus on What Matters Most

With the proliferation of AST tools that return hundreds (or even thousands) of findings with each scan, it's easy to see how results can quickly become too unwieldy to manage and triage. Even if you're using an AST orchestration solution to limit the front-end load, you will likely struggle to rationalize the disparate findings from different tests, aggregate them into a single source of truth, and prioritize them based on your organization's risk posture and policies.

It's no wonder many security and development teams struggle to answer basic questions such as

- When was my software tested?
- What was found?
- Where do my vulnerabilities come from?
- What is the extent of my exposure/exploitability?
- What was fixed?

There are three main categories of issues that prevent teams and their executives from answering these questions.

- **People.** The responsibility for AppSec is split across many teams (Development, QA, Security, etc.) and may even vary by project. Each team is often narrowly focused on its particular component or phase of the SDLC, with risk insight and prioritization influenced by this myopic approach.
- **Process.** Manual activities like code reviews and penetration tests are often not coordinated with automated testing activities. This can generate duplicate or contradictory results that add noise to the system and delay risk prioritization. This can also burden technical resources, decelerating pipelines and delaying procession toward deployment.
- **Technology.** Teams must pull findings from the multitude of AST tools they use, which categorize and prioritize findings differently. This makes it difficult to manually normalize and correlate results between them.

The solution: Application vulnerability correlation

The inability to pinpoint vulnerable software, centralize and prioritize critical findings, and track the progress of remediation efforts has led many organizations to implement an application vulnerability correlation (AVC) solution.

AVC tools provide workflow and process management capabilities that help streamline vulnerability remediation in the SDLC by normalizing AST results to a common nomenclature. They also correlate findings from myriad security testing tools and data sources in a central repository, filter out duplicate results, and assess the exploitability and severity of a vulnerability, making remediation and prioritization of security activities more effective. AVC tools optimize the triage process and reduce friction between security and development teams by automating the process flow between people, processes, and technology.

Code Dx by Synopsys

Code Dx® integrates all AST results into a centralized location and automates the most time-intensive tasks to speed up testing and remediation (see Figure 3 below).

Without automation	With automation
No bird's-eye view of results	One centralized platform to see everything
Difficult to scale AppSec with DevOps	Scale on demand
Friction between security and DevOps teas	Security and development work in harmony
Vulnerabilities found too late in SDLC	Save remediation costs by fixing earlier in SDLC
No centralized record of AppSec processes	AppSec system of record for accountability

Figure 3. The benefits of AVC tools

Code Dx helps bridge the gaps between security technologies and functional roles across DevOps workflows. It enables contributors and security practitioners to

- **Correlate results** from all your AST tools (static, dynamic, commercial, and open source) into a single console for centralized visibility and simplified security workflows
- **Prioritize vulnerabilities** using machine learning to predict which vulnerabilities are most critical to your organization and automatically send high-priority ones to your developers' issue-trackers (e.g., Jira) for remediation
- **Track remediation** activities in a system of record to manage accountability and assign tasks to specific team members
- **Unify project risk awareness** with a 360-degree view of risk for all applications (custom code, third-party components, and network) where your software resides

Code Dx fits seamlessly into the CI/CD pipeline, consolidating all your AppSec activities into a single place. Because Code Dx has two-way integrations with issue-trackers like Jira, your development team never has to interact directly with any application analyzers.

The Key to AppSec Efficiency: Application Security Orchestration and Correlation

Clearly, the rate and complexity of today's software development requires automation. This includes running the right security tools at the right time as well as managing and triaging the results. The growing adoption of security automation led Gartner to define a new category of solutions that merged AST orchestration and AVC into one: application security orchestration and correlation (ASOC).

ASOC solutions like Intelligent Orchestration and Code Dx provide the automation needed to scale security testing and identify and conduct the most impactful security activities. This enables stakeholders across security and development to keep up with DevOps pipelines, while still allowing granular control over each step of the process.

To summarize, ASOC tools

- Automate the deployment of the right tools at the right time
- Allow granular policy enforcement
- Aggregate, deduplicate, normalize, and correlate findings
- Provide audit control and reporting to support organizational and regulatory standards

Conclusion

As you look to accelerate software development, these tools will help mitigate software risk and keep your internal operations resilient. Synopsys offers a concerted approach to enable security and development teams to accomplish this.

- **Code Sight** provides developers with a quick analysis of source code and open source, with remediation guidance to establish software security before assets are pushed downstream.
- **Seeker** provides a side-stream mechanism for interactive application security testing that occurs concurrently with other functional and security testing, verifying risks and extending security to APIs, microservices, and other supporting technologies.
- **Intelligent Orchestration and Code Dx** comprise a complete ASOC solution that enables organizations to establish an AppSec system of record, coordinate testing intelligently, and gauge their most impactful security activities based on risk.

Learn more about our solutions for DevSecOps and request a demo today.

1 Terry Ray, Billions of Compromised Records and Counting: Why the Application Layer is Still the Front Door for Data Breaches, Threatpost.com, June 8, 2021.

2 IBM.com, Cost of a Data Breach Report 2022, IBM, 2022.

3 Dave Gruber, Cracking the Code of DevSecOps, ESG, June 2021.

4 Ibid.

5 Ayal Tirosh, Hype Cycle for Application Security, 2017, Gartner, July 28, 2017.

SYNOPSYS®

The Synopsys difference

Synopsys provides integrated solutions that transform the way you build and deliver software, accelerating innovation while addressing business risk. With Synopsys, your developers can secure code as fast as they write it. Your development and DevSecOps teams can automate testing within development pipelines without compromising velocity. And your security teams can proactively manage risk and focus remediation efforts on what matters most to your organization. Our unmatched expertise helps you plan and execute any security initiative. Only Synopsys offers everything you need to build trust in your software.

For more information, go to www.synopsys.com/software.

Synopsys, Inc.
690 E Middlefield Road
Mountain View, CA 94043 USA

Contact us:
U.S. Sales: 800.873.8193
International Sales: +1 415.321.5237
Email: sig-info@synopsys.com