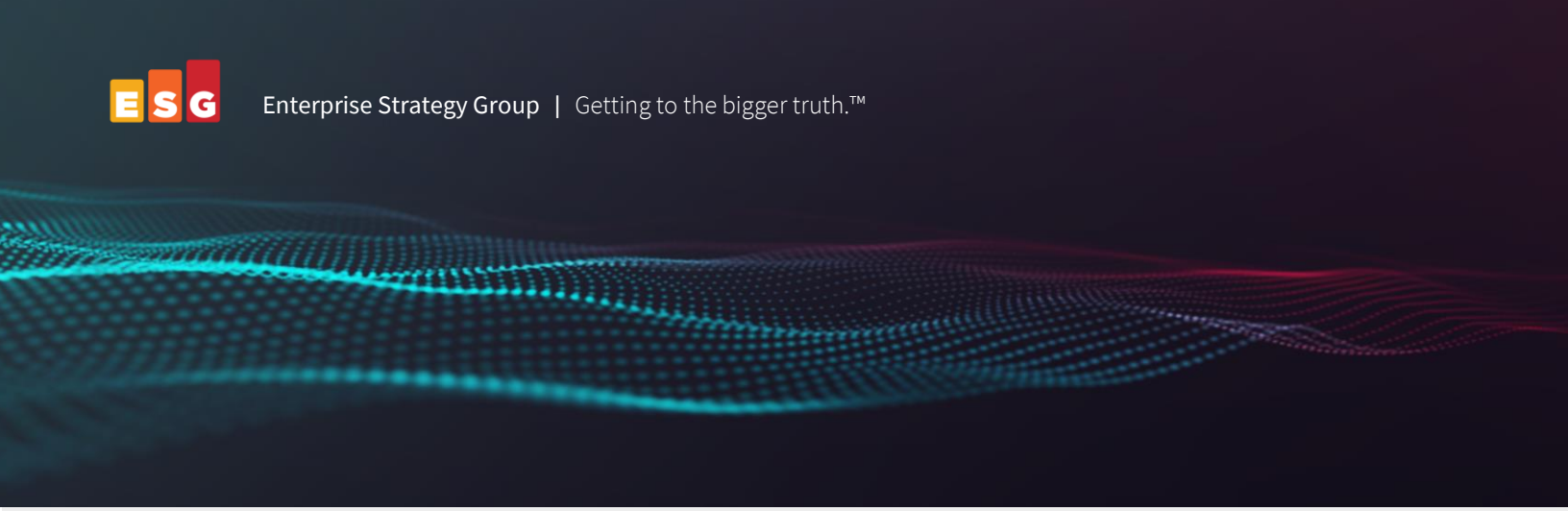


ESG WHITE PAPER

Cracking the Code of DevSecOps

Achieving Velocity Requires a Modernized Approach to Application Security

By Dave Gruber, ESG Senior Analyst

June 2021

This ESG White Paper was commissioned by Synopsys and is distributed under license from ESG.

Contents

Executive Summary

Application development practices continue to evolve, enabling development teams to deliver applications at a pace never thought possible. At the same time, cyber-criminals have developed new levels of attack strategies while intensifying their focus, making it more important than ever to scrutinize applications for security vulnerabilities.

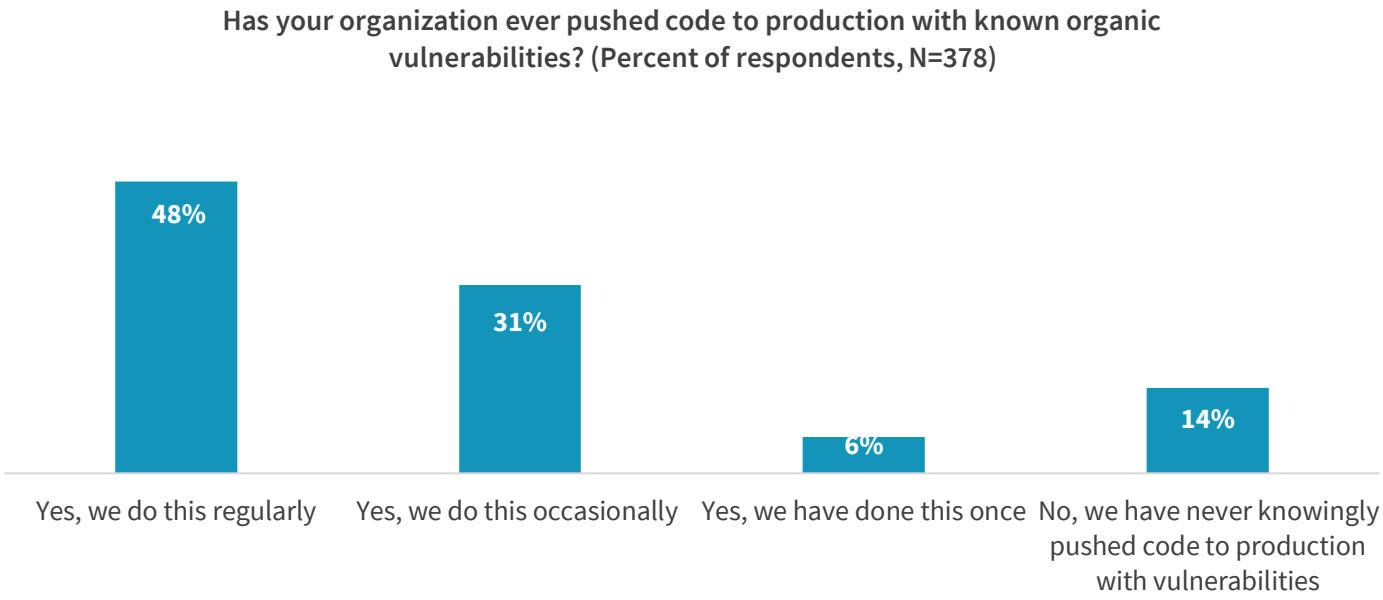
Development and security teams are responding by shifting security further left and investing in tools integrations. Many believe that improved DevOps integration is the answer, with 43% of ESG research respondents reporting that it is one of the most important things they can do to improve their application security programs.[1]

58% of organizations say that application security is their top security investment priority.[2] Yet while organizations continue to invest in application security, they have many challenges to overcome. Developers often lack the knowledge to mitigate issues. Integration between different application security tools is difficult. And the added friction caused by security tools is slowing down development velocity.

With digital transformation initiatives continuing to accelerate, development teams are forced to make tough decisions between meeting time-to-market objectives and mitigating risk. Despite ongoing investments in application security programs, 79% of organizations admit to pushing application changes with known vulnerabilities (see Figure 1). When asked why, 54% of organizations report that the need to meet critical deadlines forces teams to make decisions to move forward with vulnerable code, leaving organizations at risk.

Current security strategies are simply not scaling to keep up with modern development practices. Change is needed.

Figure 1. Despite Investments in AppSec, Nearly Half Regularly Push Vulnerable Code



Source: Enterprise Strategy Group

[1] Source: ESG Research Report, *Securing Modern Application Development Environments*, December 2020. All ESG research references and charts in this white paper have been taken from this research report, unless otherwise noted.
[2] Source: ESG Master Survey Results, *Modern Application Development Security*, November 2020.

Digital Transformation is Driving Development Velocity

As digital transformation initiatives continue to reinvigorate businesses, software applications have become an anchor for customer and commerce interactions. Business leaders need both speed and agility to compete in this fast-moving digital economy, putting pressure on development teams to deliver. Development teams are responding by investing in cloud-native architectures together with agile development practices and DevOps/GitOps automation to accelerate development velocity.

Software Complexity and Development Velocity are Rapidly Increasing

Modern software is more complex (e.g., microservices & serverless), delivered faster (e.g., 10s -> 100s -> 1000s of builds per day), and deployed in more ways than ever before (e.g., containers, cloud, IoT devices).

Deployment is increasingly automated, leveraging DevOps toolchains and practices that enable faster development and delivery.

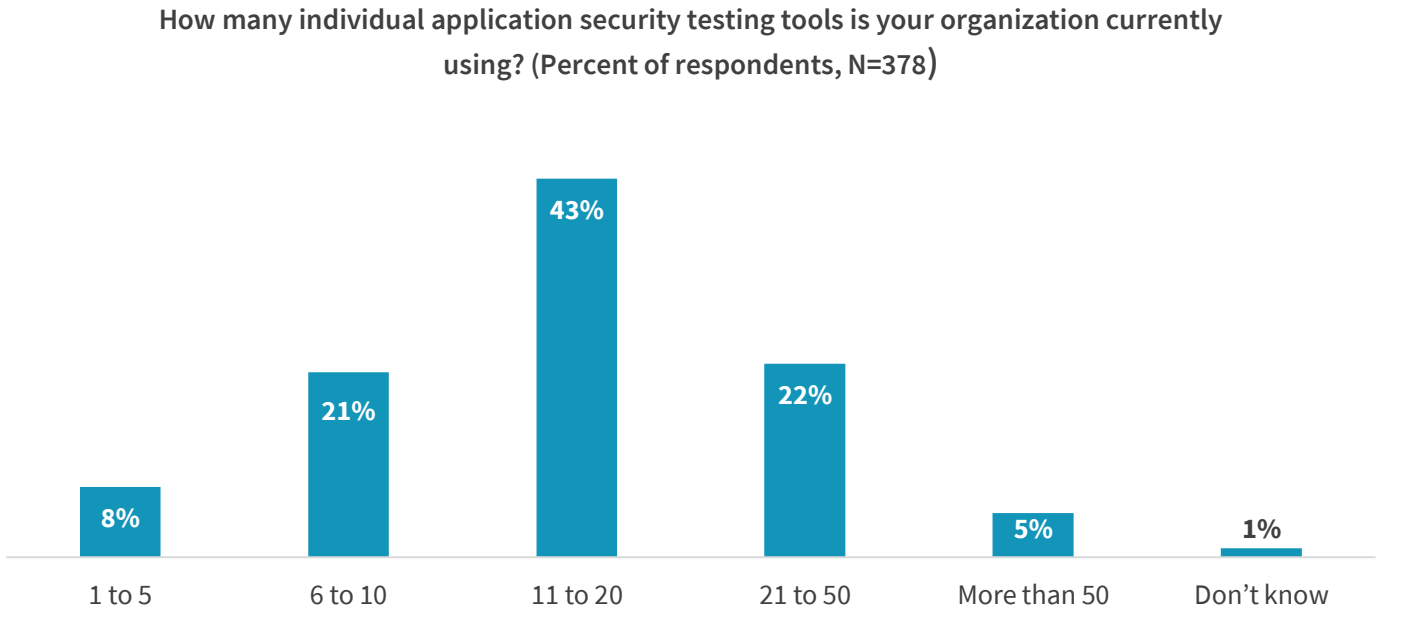
While these strategies are enabling development teams to deliver faster, software complexity is on the rise, adding new challenges for application security teams.

Securing Modern Applications

Modern application security programs are a complex mix of manual and automated tools and practices. Application vulnerabilities are a top cybersecurity risk, motivating investment in AppSec-specific security resources, automated testing tools for vulnerability assessment, and developer security training programs. 71% report the use of application security tools on more than 50% of their code base, with 69% rating the effectiveness of their program as an eight or higher on a scale of one to ten.

More than two-thirds of ESG research respondents report the use of 11 or more automated application security testing tools (see Figure 2), including tools like SAST, DAST, IAST, fuzz testing, and container scans. Many also employ manual testing activities including pen testing, code reviews, threat modeling, and red teaming to further secure their application environment. To keep up with accelerating development cycles, security teams are working to shift security left to uncover issues earlier in the development process. Investments in DevOps integration automate much of the assessment process and add consistency. However, manual triage and prioritization by development and security analysts is still required.

Figure 2. Individual Application Security Tools in Use



Source: Enterprise Strategy Group

Challenges

Despite ongoing investments, many organizations, especially those managing large application portfolios, are struggling to implement and operate effective application security programs. While many would agree that integrating and automating security testing within development toolchains and workflows is essential, it all too often slows development pipelines due to a mismatch between the velocity of build and security testing activities. While development strategies rapidly advance, application security strategies are lagging, putting businesses at risk.

ESG research identified several key challenges with current application security strategies, including:

- Too Many Findings - A single scan by an application security testing (AST) tool can surface 100s or even 1,000s of findings. Teams that integrate and automate full AST scans in their CI pipelines often find that the sheer volume and duplication of results from tests performed at different stages of the SDLC is a problem. Even though only a small percentage of the findings may present enough risk to require immediate attention, identifying and prioritizing high-priority vulnerabilities is a significant burden. Development and security leaders are forced to triage and prioritize issues for remediation, often requiring difficult decisions about whether to prioritize delivery schedules over application security.

- Proliferation of Tools and Scans – 30% say that they are overwhelmed by the number of testing tools in use. Further, 26% of organizations say that their collective application security tools are adding friction to their development processes, impeding development velocity. Running multiple testing tools at different points in the SDLC can produce duplicate results that need to be correlated and deduplicated later. Correlating and prioritizing findings across multiple AST tools is challenging and time-consuming, requiring involvement by both security and development leaders. Most fail to effectively stitch together related findings, further increasing the backlog of remediation activities.

- Lengthy Scan Cycles - As application development velocity continues to accelerate, traditional approaches to application security are failing to scale. While build pipelines are often intended to run in seconds to a few minutes,

AppSec tool scans can often take several minutes or even hours. This problem is compounded because multiple forms of analysis (e.g., SAST, SCA, etc.) must often be performed. As a result, teams find that integrating AppSec into their pipelines disrupts velocity goals.

- Poorly Aligned Risk Models – Security tools are all too often applied across all application changes, regardless of their risk profile. When organizations lack clear policies on what assessment tools are needed for different risk scenarios, broad-brush approaches are employed, resulting in inadequate testing for high-risk applications, while wasting time and resources on low-risk application changes. While AST tools often include policy enforcement and vulnerability reporting capabilities, they are generally siloed implementations, typically applied at different points in the SDLC. This siloed approach makes it difficult for teams to implement policies, identify the highest priority software security risks, and aggregate reporting across the multiple AST tools in use.

- Disconnected Security Activities – Automated testing tools are good at identifying vulnerabilities in code and configuration; however, they often fall short identifying larger, architectural issues that cause security problems. To fill these gaps, most organizations employ one or more manual security testing practices, including threat modeling, code reviews, and penetration testing. These activities are often initiated by development and security leaders without alignment with risk policies. This makes it difficult for security teams to implement consistent and timely application security governance that is aligned with development activities, and equally difficult to harmonize findings from manual activities with those produced by automated tools.

- Too Many Exploits - Despite the use of multiple application security tools, 81% of organizations still report that they've had applications exploited, with 60% reporting that they've had applications exploited by OWASP Top-10 vulnerabilities within the previous 12 months.

What's Needed

It's clear that simply integrating and automating security testing tools in CI pipelines to test everything all the time doesn't scale to meet the demands of modern application development. A new model for application security is required. Simply stated, software security is impeding DevOps velocity. **Organizations need to modernize their approach.**

- Application security strategies must embrace a new, risk-driven, security-where-needed approach, focusing more stringent controls on higher-risk application changes while backing off security testing in lower-risk areas. These risk-driven security controls must intelligently and seamlessly integrate with core DevOps without interfering with development velocity, while enabling development and security teams to align around and achieve security objectives.

- The solution should factor in individual application risk profiles and align them to security policies and profiles to determine what (and when) automated and manual controls should be applied.

- Security teams should define automated rule sets that govern how risk is managed, enabling a more intelligent, automated orchestration process for test execution, policy enforcement, and issue prioritization and filtering.

- This process should operate independently of the core DevOps pipeline, enabling security and development teams to operate independently. Development teams must

DevOps Integration is Critical, but AppSec Must Not Add Friction to the Process

Development teams must have the flexibility to leverage advanced development techniques without added friction from additional security controls.

have the flexibility to leverage advanced development techniques without added friction from additional security controls.

- Optimized for personas, the solution should provide security analysts with the right tools and information to assess risk, while providing developers with the right info and tools to mitigate risk.

Intelligent Orchestration and Correlation Solutions from Synopsys: A Modern Approach to AppSec

Application Security Orchestration and Correlation solutions from Synopsys help teams modernize their AppSec programs to match the velocity and scale of their application development and DevOps workflows. Synopsys Intelligent Orchestration is a dedicated Application Security CI pipeline that optimizes AppSec testing while removing complexity from DevOps toolchains. It runs in parallel to existing build and release pipelines to perform necessary application security tests based on defined policies, application profiles, and SDLC events. Code Dx leverages machine learning to aggregate, correlate, and prioritize risk across multiple ASTs, reducing noise while helping development and security teams identify and remediate the highest risk issues. Together, they address both the pipeline friction and vulnerability overload that hinder many DevSecOps programs.

- Integrate easily with your existing pipelines and development toolchains.
  Intelligent Orchestration and Code Dx connect to DevOps pipelines with a couple of simple API calls, eliminating the need to reimplement build and release pipelines to add security testing. A rich and extensible set of AppSec and DevOps integrations enable further integration with a wide range of development, security, and issue tracking tools already in use.

- Ensure the right tests are run at the right time.
  AppSec policies can be defined as code, specifying rules for security evaluation, response, and notification. Using proprietary technology, Intelligent Orchestration applies those rules to code changes and other SDLC events to trigger relevant, appropriate security tests. This intelligent approach maximizes velocity by performing only the tests that are needed at the time they are needed.

- Deliver the right information to the right teams.
  Code Dx provides optimized and standardized reporting of application risk insights across multiple AppSec tools. Findings are automatically correlated, filtered, and prioritized based on risk and delivered to developers directly within development and defect tracking tools, avoiding vulnerability overload and enabling teams to achieve the maximum risk impact at minimal cost.

- Automate the workflow for manual or out-of-band AppSec activities.
  Intelligent Orchestration can be configured to trigger manual AppSec activities, such as penetration tests, through existing defect tracking systems and communication channels, based on policy. This enables security teams to coordinate security compliance workflows with development workflows and SDLC events.

- Aggregate and correlate findings from multiple testing tools.
  Code Dx aggregates and correlates findings from Synopsys, open source, and other third-party application security tools, reducing noise, highlighting findings that present real business risk, and providing insight into efficacy of the tools in use. Using machine learning, Code Dx automates resource- and time-intensive triage and prioritization, helping security and development teams focus on the highest areas of risk.

- Centralize risk visibility.
  Code Dx provides a consolidated view of application security activities and software security risks at the project or business unit level, tracking what software was tested, what risks were found, and if/when risk was remediated.

The Bigger Truth

Security teams have been under tremendous pressure to "shift left" and integrate their tools and processes with the increasingly automated world of software development. Unfortunately, they often find that traditional application security strategies fail to scale with modern development practices. As application development velocity plays an increasingly important role in the digital business competitive landscape, new security strategies must be implemented to minimize risk. Without rapid change, risk will escalate quickly as development teams make tradeoffs between time to delivery and security.

DevOps must be freed from the friction of application security, enabling development teams to explore and embrace technology advances that lead to increases in velocity and scale. Concurrently, security teams need the ability to define risk-driven policies that can be applied to aggregated, correlated findings from multiple ASTs. This highly orchestrated, automated approach can free DevOps from the friction caused by application security, while enabling security teams to more efficiently identify and mitigate areas of high risk.

Advances in application security solutions from vendors like Synopsys are enabling the modernization of application security programs, enabling businesses to accelerate digital transformation initiatives while managing associated risk.

Enterprise Strategy Group is an IT analyst, research, validation, and strategy firm that provides market intelligence and actionable insight to the global IT community.


www.esg-global.com


contact@esg-global.com


508.482.0188