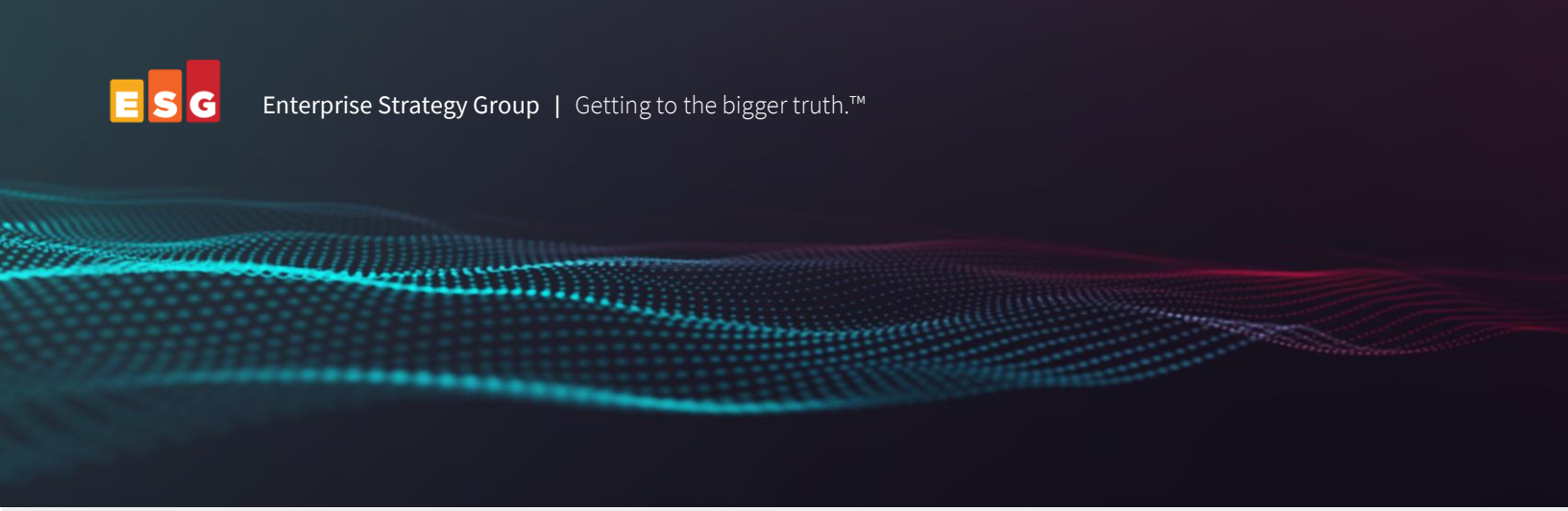


ESG WHITE PAPER

Modern Application Security is Failing

A Multidimensional Approach to Application Risk with Apiiro

By Dave Gruber, ESG Senior Analyst

May 2021

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

Contents

Application Security Programs are Failing to Scale

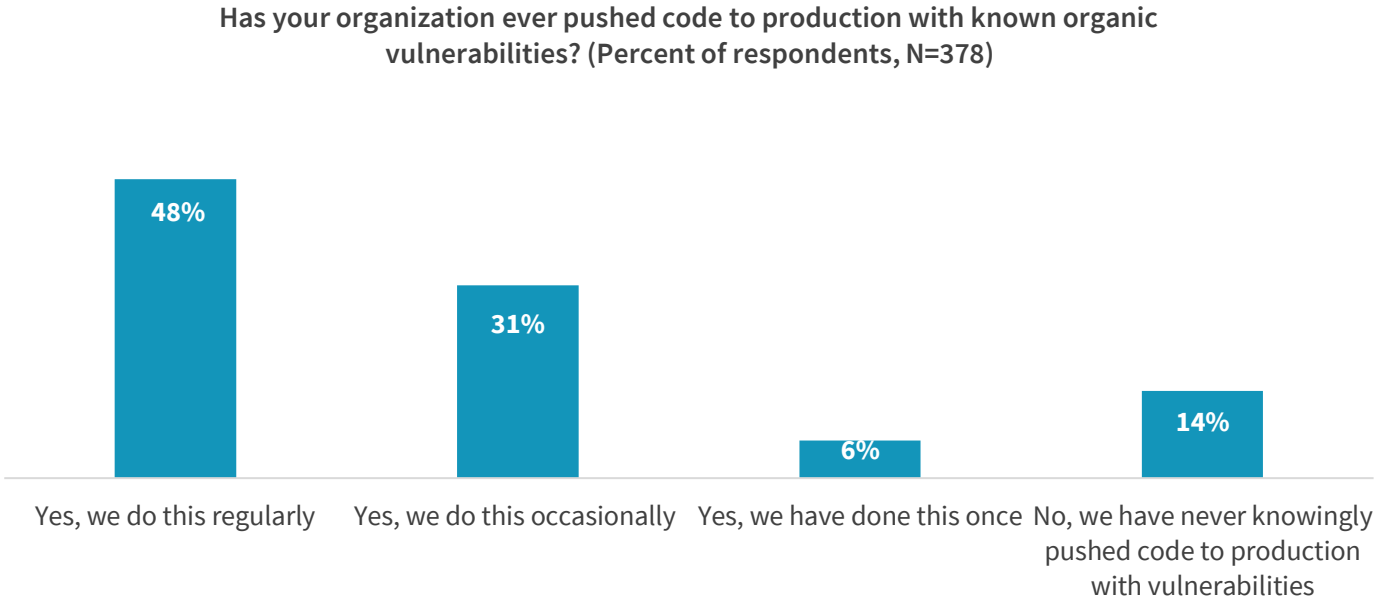
Application development teams are under unprecedented levels of pressure to deliver as businesses accelerate digital transformation initiatives. As AppDev teams accelerate code development through code reuse, open source, cloud platform advancements, and investments in DevOps, Application Security teams are struggling to implement tools that can ensure high levels of application security while aligning with development velocity objectives. Modern deployment models are further adding complexity to application development risk assessment, requiring a combined analysis of application code changes, infrastructure code changes, API gateway changes, and cloud configuration changes to effectively determine material risk.

Development teams knowingly push application updates with risks on a regular basis. ESG research tells us that 79% of organizations knowingly push vulnerable code—48% regularly and 31% occasionally. When asked why, 54% said they did so to meet a critical deadline, with a plan to remediate in a future release.[1] These teams are compromising security by releasing according to business demand.

Despite the use of multiple automated security tools, security, development, and compliance teams spend too much time focusing on the wrong things. With most tools lacking the necessary context to understand the true risk of a change, teams end up focusing too much time on changes with little business impact, and not enough time on the changes that matter. While multiple SQL injection vulnerabilities may introduce significant risk to an internet-facing application, the same vulnerabilities could be classified as low risk for an internally facing application. Accurate risk assessment requires context beyond simple vulnerability severity and application risk profiling. Understanding the likelihood and potential impact of an issue can completely change risk decisions.

So, the real question is, given the limited nature of AppSec tools and manpower, how effective are organizations at understanding the risks associated with pushing application changes with known risks, such as vulnerabilities, exposing PII, or the use of an unauthenticated API? Are these organizations truly making informed decisions?

Figure 1. Organizations Knowingly Push Vulnerable Changes



Source: Enterprise Strategy Group

[1] Source: ESG Research Report, *Securing Modern Application Development Environments*, December 2020.

Figure 2. Time to Delivery Often Trumps Security

Source: Enterprise Strategy Group

Shifting Security Left Is Not Enough

Despite the use of automated testing and analysis tools and the drive to move security left, the process of issue triage and prioritization is still highly manual and time-consuming. Existing tools often lack the necessary context for effective risk scoring, causing many development teams to waste time remediating issues in code with little material risk and minimal real-world business impact, while code that introduces material risk often ships lacking appropriate security scrutiny.

Further, 39% of development teams depend only on individual developers or development managers to prioritize and assess the severity of issues.[2] When manual risk decisions are left to individual managers, results are inconsistent, unpredictable, and lack auditability.

While 43% of application security teams believe that improved DevOps integration is the most important thing to do to improve their application security program,[3] there are fundamental issues in the current AppSec model that prevent the process from scaling to keep up with development velocity. These issues slow application delivery and reduce the efficacy of security program investments.

Challenges

Accelerating Code Development Delivery

While development organizations are successfully accelerating development cycles through investments in DevOps automation, the application security process lacks similar refinements, leading to the generation of more security alerts, many of which have little to no real-world business impact. This situation creates additional headwinds for development teams who are already under pressure to deliver faster. With developers releasing faster and a median ratio of full-time security resources to developer resources of 159 to 1,[4] the current application security model is failing to scale.

[2] Ibid.

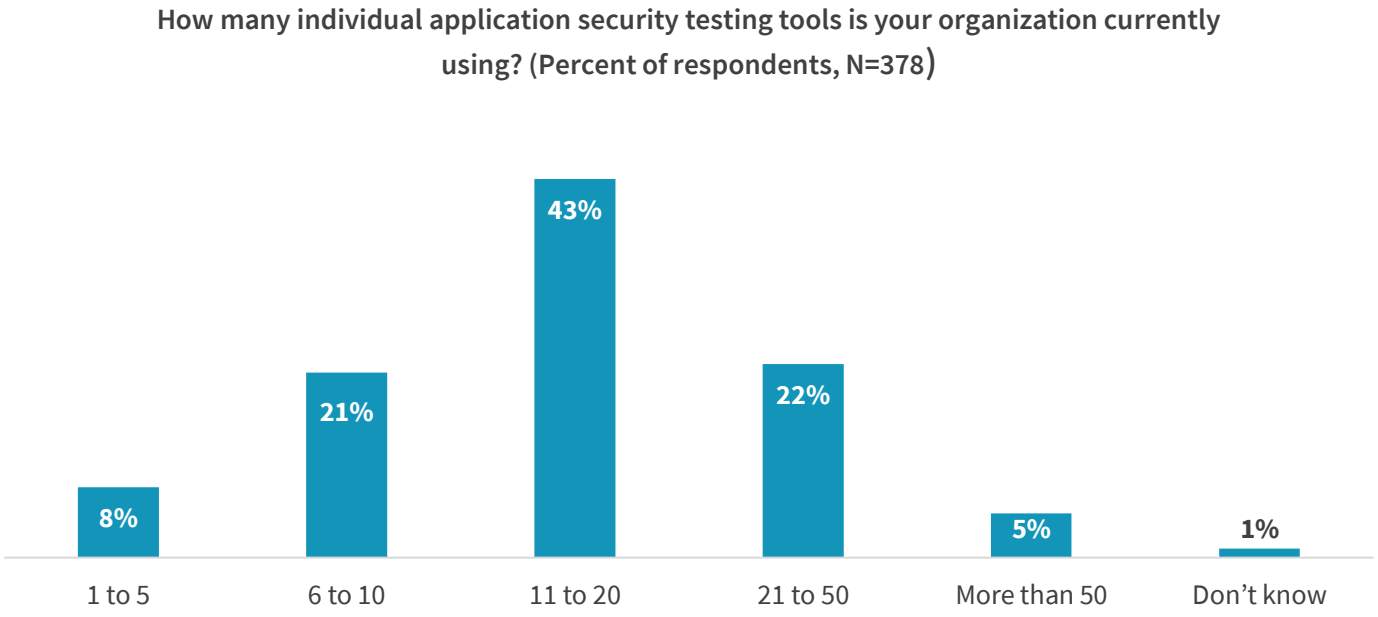
[3] Ibid.

[4] Source: *Building Security in Maturity Model (BSIMM) – Version 11*.

Too Many Alerts with Too Little Impact

With 70% of organizations reporting having more than 10 application security tools in use,[5] AppSec teams are buried in alerts and struggling to maintain development velocity while triaging and remediating issues. Without the necessary context to automate the triage process, AppSec analysts and dev managers spend inordinate amounts of time sifting through alerts to determine where to prioritize remediation activities.

Figure 3. The Need for More Testing Tools Results in Too Many Alerts



Source: Enterprise Strategy Group

Inconsistent, Highly Manual Risk Assessment Processes

As security analysts and dev managers sift through mountains of alerts, they are mired in assessment and prioritization of remediation actions based on their own contextual knowledge about applications, specific code/feature modifications and additions, experience with the individual developers who created them, and the overall risk profile of the applications they power. This highly manual but critical analysis and prioritization process drives key decisions about what code ships with known vulnerabilities or other unknown risk, and what code does not, directly impacting the risk profile of application changes.

Lack of Context

Most automated tools lack the necessary context to assess risk associated with individual user stories and associated pull/merge requests (PR/MR). To fully understand the context of changes, organizations must pay close attention to code throughout the history and in a continuous manner across repositories, and enrich it with the contributors' knowledge, commit message, pull request discussions, and ticketing system data. Assessing changes at just a current point in time fails to identify risk associated with the application.

[5] Source: ESG Research Report, *Securing Modern Application Development Environments*, December 2020.

Without a contextual understanding about specific application functions, where an application is deployed, what data or service a specific function has access to, whether data is read-only versus updatable, or what developer expertise is involved in code development, development and security managers lack key insights required to properly assess risk. These and many more contextual attributes impact risk assessment and the associated security scrutiny required to ensure proper risk mitigation and management.

Without continuous risk classification throughout the development process, user stories and associated pull/merge requests (PR/MR) are often all treated equally (like spreading peanut butter evenly across them all), with organizations applying expensive security analysis processes and development remediation across all PRs/MRs, regardless of material risk. However, all code and applications are not created equally. When security analysts and developers waste time on assessing and remediating changes that involve minimal material risk, they end up spending less time on the risks that really matter, often missing risky material changes or vulnerabilities with high business impact. Risk varies based on whether the changes are front-end or back-end, whether the functions are public-facing or internally facing, whether data is sensitive or non-sensitive, whether transactions are read-only or read and write, and more.

Current Strategies Are Not Keeping Up

Shift-left

Speeding Up Existing Processes is Insufficient

The fundamental issue is the lack of automation for assessing business risk, not prioritizing vulnerability severity. It is not about which vulnerabilities to focus on; it's about what application changes to focus on.

Modern AppSec programs continue to focus on shifting security as far left as possible to remediate issues early, reducing the higher costs of downstream remediation. While this strategy helps remediate issues sooner, it ignores the broader issue of assessing whether issues are worth remediating based on their risk impact. So, while shifting left makes sense to minimize remediation costs, without evaluating context and potential business impact within the process, time and resources are often wasted remediating low-risk issues.

More DevOps Automation and Integration

While 43% of organizations believe better DevOps integration is one of the most important things they can do to improve their application security program,[6] there are logical limits to the benefits of this automation. Ultimately, remediation falls on developers, so even with more DevOps automation, low-impact, poorly prioritized issues with low business risk negatively impact developer productivity. Simply bolting on existing tools and processes to automated DevOps processes is not working.

AppSec Orchestration

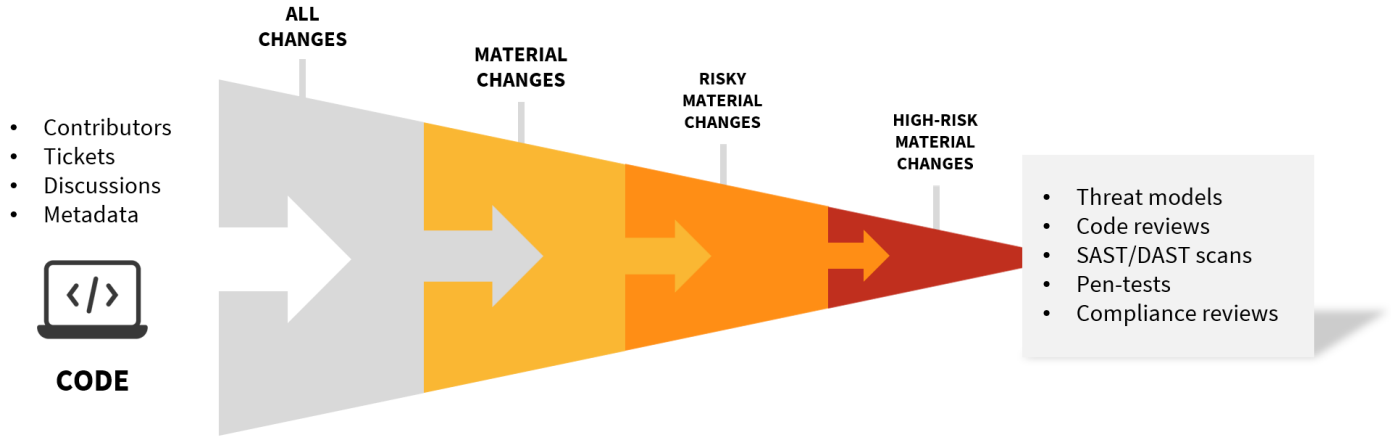
Many are applying orchestration layers that sit on top of multiple application security tools. However, these tools alone cannot address many of the fundamental challenges. While orchestration tools can aggregate and correlate alerts, they lack the ability to understand enough context to assess business impact.

[6] Source: ESG Research Report, *Securing Modern Application Development Environments*, December 2020.

A Multidimensional Approach to Application Security

When a multidimensional approach to application security is used, security controls can be optimized to closely align with the potential business risk associated with individual pull/merge requests and associated user stories. This helps development teams free up valuable resources, speed development, and redirect precious security resources to further scrutinize security within high-risk application areas.

Figure 4. The Risky Material Change Funnel



Source: Enterprise Strategy Group

Core to a risk-driven application security approach is the ability to automate a contextual risk assessment. Context includes the combined risk associated with code, configuration, data, developer experience, and potential business impact. This context is absent in most application security tools, requiring either the manual addition of context by security and development managers, or more often, mis-prioritized issue alerts because of the lack of context.

Because only a small percentage of changes introduce real, material impact, there is a significant opportunity to reduce the amount of security scrutiny for low-risk changes, redirecting precious resources to high-risk, material changes. With the proper context, many organizations may find that risky material changes are less than 1% of their total commits.

When automated, contextual risk assessment is integrated with the DevOps pipeline and applied continuously throughout the SDLC (from design, through code, test, and production), automated decisions can be made about what level of security controls are necessary to minimize material impact.

Automated remediation playbooks can be created and automatically initiated against cohorts of risky changes that share similar risk profiles. Remediation actions can include everything from peer-code-reviews through SAST/DAST/IAST analysis, to penetration testing, and more. In many cases, multiple controls will be necessary. This automated approach enables security and development leaders to adjust security controls based on material business risk assessment.

With this new level of risk fidelity and continuous security assessment, review and controls can be applied to those application changes that potentially introduce the most material risk, ensuring the most stringent security assessment, review, and remediation, while additionally satisfying risk and governance requirements. Conversely, low-impact, low-risk application changes should require less stringent controls, freeing up valuable development and security time and resources that lead to accelerated velocity.

Figure 5. Aligning Security Controls to Material Business Risk

Source: Enterprise Strategy Group

Contextual Risk Assessment

Every user story and associated individual pull/merge request has a potentially different risk profile. Accurate risk assessment requires broad context, including the assessment of code, API configuration and consumption, cloud/deployment configuration, what data is accessed or modified, and the specific experience and skills of the developers involved. Automating this assessment process ensures consistency and scrutiny of modifications that have the most material business impact.

Context Matters

When automated risk assessment processes include context across developers, code, and deployment environments, risky material changes can be automatically prioritized along with the creation of an actionable work plan. By focusing on the risks that matter, organizations' security experts can better prioritize and remediate the true threats to their business.

Individual developers possess a multitude of skills and experience, yet they often find themselves working on code outside their skill zone, potentially introducing additional risk. In these scenarios, additional security scrutiny is often required, leveraging code reviews or other automated assessment tools. When automated tools can factor in developer skills and experiences, they can trigger additional security processes or tools to mitigate risk. Simply automatically assigning a security champion within the development team to review the new code can make a security champion program better and more efficient.

Automated Risk Classification and Security Policies

Automating the process of assigning individual application changes to risk categories enables security teams to craft specific security policies and testing "playbooks" that govern what security controls are required for different risk profiles. Security policies applied to high-risk changes may include a combination of automated testing tools, such as SAST, DAST, and SCA, together with manual code review and potentially pen-testing in advance of deployment. Policies crafted for low-risk changes, such as a UI change to a navigation screen, may conversely require very little security scrutiny.

This flexible, dynamic application security strategy enables both development and security teams to optimize security and remediation activities based on the material risk associated with individual application changes.

Alignment of Security Controls

Combining automated risk assessment, risk classification, and the application of highly customized security policies enables development and security teams to adjust security scrutiny based on the material risk associated with individual application changes.

When automated tools can assess the individual potential risk associated with a PR/MR, and can trigger well-aligned security workflows, development velocity increases. This critical alignment process transforms application security into a dynamic and variable process, highly tuned around the risk assessment of individual application changes.

Introducing Apiiro: Multidimensional Application Risk Management

Apiiro is reinventing the secure development lifecycle by providing complete risk visibility and control with every change, from design to code to cloud. Whether or not organizations have mature application and cloud security programs in place, Apiiro helps manage all risks.

Gain Complete Application Inventory & Risk Visibility

- Get a 360° comprehensive risk view across applications, infrastructure, developer knowledge, and business impact.

Manage the Risks that Matter

- Prioritize and manage risks across the SDLC process, from design to production.

Enforce Compliance & Governance

- Automatically prevent security & compliance violations before CI/CD.

Automate Remediation Workflows

- Trigger automated workflows to remediate new risks.

Prevent Advanced CI/CD Attacks

- Detect and respond to build-time attacks and compromised developer identities.

The Bigger Truth

Despite the use of automated security testing and analysis tools, development organizations continue to struggle to secure applications. As automated DevOps processes are helping development teams deliver faster, the people responsible for making sure applications are secure are drowning.

Most automated security tools lack the necessary context for effective risk scoring, requiring dev and security leaders to utilize a highly manual, inconsistent triage and prioritization process, resulting in the wasting of precious development time remediating issues in code with little material risk, while code that introduces material risk often ships lacking appropriate controls. The lack of automated, risk-driven application security lowers efficacy and adds unnecessary remediation investments, slowing development velocity.

When automated tools can accurately assess the potential risk associated with an individual PR/MR, security controls can be aligned to risk profiles, increasing scrutiny in higher-risk areas while decreasing unnecessary security processes in lower-risk areas. With this new level of risk fidelity, security assessment, review, and controls can be applied to those code changes that potentially introduce the most material risk, ensuring that the most stringent security assessment, review, and remediation takes place on that code. Conversely, code changes associated with low-impact, low-risk feature sets require less stringent controls, freeing up valuable time and resources that lead to accelerated velocity.

Organizations who are looking to overcome challenges in modern application security programs should consider solutions like the Apiiro Code Risk Platform.

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