ESG SHOWCASE

Transforming the Role of Security Analyst From Gatekeeper to Developer Enabler

Date: December 2021 **Author:** Melinda Marks, Senior Analyst

ABSTRACT: As organizations adopt modern software development practices for greater speed and agility, application security is lagging behind with outdated approaches based on waterfall development processes. Application security programs are reactive, finding issues after they have been pushed to production, and security analysts spend a majority of their time sifting through security findings, triaging and prioritizing remediation efforts, and contacting developers for remediation work. Organizations need a modernized approach that aligns application security with agile software development processes, driving both speed and improved security posture.

Development Demands a Modern Security Approach

Traditional application security was once well aligned with infrastructure provisioning and common waterfall development processes. An application developer would file a ticket, wait for someone from IT or operations teams to provision a server, work to release software, and then issue updates over periods of months or even yearly. For security, it was a matter of running vulnerability scans and impact assessments at set points along this waterfall process.

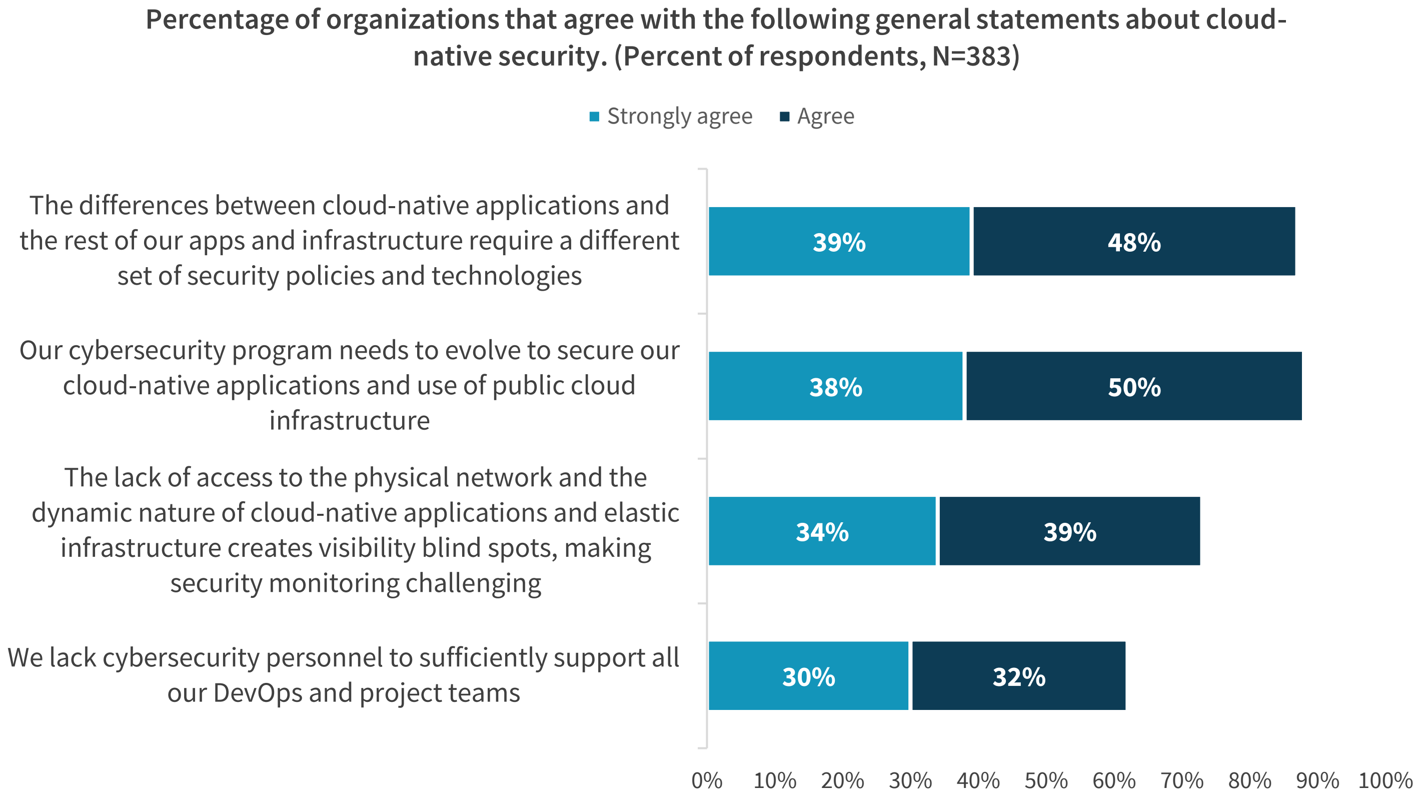
Today, organizations are moving to cloud-native application development, where developers use cloud platforms, like AWS, Google, and Azure, to provision infrastructure with minimal help from other teams. They can release and update software using continuous integration and continuous deployment (CI/CD) for rapid software releases and updates. When developers are continuously releasing and updating software, trying to insert traditional application security processes can slow things down, and security can't scale to keep up with the speed of agile development.

If developers feel that integrating security processes or interacting with security will slow them down, security is often left out of the development process. Security testing in development can be disruptive, and while monitoring applications in runtime can detect issues, it's a reactive strategy; when an issue is found, the security analyst has to prioritize the issue, track it back to the owner, and then work with the developer to fix it.

According to ESG research, 88% of respondents agree or strongly agree that their cybersecurity program needs to be updated to more effectively secure their cloud-native applications and use of public cloud infrastructure (see Figure 1).[1] Almost three-quarters (73%) recognize that the lack of access to the physical network and the dynamic nature of cloud-native applications and elastic infrastructure create visibility blind spots, making security monitoring challenging.

[1] Source: ESG Research Report, *The Maturation of Cloud-native Security*, May 2021. All ESG research references and charts in this Showcase have been taken from this research report, unless otherwise noted.

Figure 1. Many Organizations Still Need to Evolve Security Strategies to Accommodate Cloud


Percentage of organizations that agree with the following general statements about cloud-native security. (Percent of respondents, N=383)

Source: Enterprise Strategy Group

Working With Instead of Against Developers

It is important for security to work with developers since they share a common goal: to ensure that applications are shipped secure, without coding defects. A modern application security approach helps security work with developers to efficiently detect and fix any coding issues, shortening the feedback loop and enabling developers to ship their code with the confidence that it is secure.

When security flaws can be fixed early in development before the product ships, the core metrics for the security analysts can shift from quantities or percentages of vulnerabilities mitigated to acceleration of mean time to remediate (MTTR).

Contextual Awareness of Application Execution

To be effective, it's important for developers to have context on how the applications function and how they are used. This helps them to fix definitive issues that can be exploited, instead of wasting time on suspicious findings that may not matter.

Over the years, there has been a lot of talk about shift-left strategies, but we haven't been able to shift left *securely* with outdated application security tools. With the right solution, we can shift left and trust developers to secure the code they are building and designing. After all, it is in their interest to deliver secure, high-quality code that meets functional, performance, and security requirements, from design and build to test and deploy.

When security becomes an attribute of software quality instead of a competing process, developers can focus on the rapid, agile delivery of features without worrying about out-of-band security processes slowing them down. And security gains the confidence that developers can take responsibility for securing their code without their help.

Accuracy Matters

A modern application security approach requires embedding security processes—such as testing, analysis, and threat detection—across the development lifecycle. The accuracy of these processes is the key to being able to securely shift left. When the solution can accurately surface security issues and how to fix them, developers can fix problems themselves early so fewer problems materialize in production.

With fewer security issues in production and fewer problems to triage, security analysts can shift their focus to services that help facilitate secure development, including automating testing processes, setting up standards or policies as guardrails, and coaching development teams.

A successful program includes assistive automation throughout the software development lifecycle (SDLC), optimized for each stage: design, build, test, and deploy. When mechanisms are architected and fully integrated to assist development, QA, and DevOps engineers in meeting their objectives, security can scale to grow and mature with modern development.

A New Paradigm

This approach transforms security from gatekeepers reacting to security vulnerabilities in production to developer enablers setting up policy and automation so development can efficiently release secure code. With this paradigm shift

1. Instead of forcing developers to learn about security, embedded security processes give them info on coding issues so they can fix them early with assistive tools for making the needed changes.
2. Developers see security as helpful for providing guardrails and automated processes that ensure delivery milestones are met with quality, performance, security, and reliability.
3. Feedback loops are shortened so security can work with the software development lifecycle instead of against it.

Remediation Matters Most

Modern application security program success metrics need to lead with MTTR indicators focused on issue resolution instead of issue detection. When security focus is moved to shipping secure applications, there are fewer issues to detect.

Introducing The Contrast Application Security Platform

The Contrast Application Security Platform provides a modernized approach that works across the software development lifecycle, enabling security to scale with agile and cloud-native development. It enables full observability from a single source of truth with automated mechanisms for finding and correcting security issues across the lifecycle.

It includes:

- One platform for centralized visibility and observability across distributed systems. This gives multiple teams—developers, security team members, and cloud operations teams—a unified source of truth with robust reporting to manage security posture and document compliance.
- Continuous serverless application security monitoring by testing for vulnerabilities in cloud-native environments like AWS Lambda.

- Automated code testing and analysis embedded in SDLC processes, including automated software composition analysis (SCA), pipeline-native static application security testing (SAST), and continuous vulnerability assessments with interactive application security testing (IAST), greatly reducing MTTR compared to other solutions.
- Developer self-service solution focused on finding exploitable flaws early within native CI/CD workflows, leveraging actionable "how-to-fix" remediation guidance. It includes the ability to block known and unknown application attacks detected in runtime.

Instrumentation for Accuracy

Contrast Security's platform embeds security throughout the software development lifecycle by using instrumentation with sensors embedded in the application code. The sensors continuously monitor the applications with visibility into custom and third-party libraries including transitive dependencies. Where Contrast differs from other tools is its ability to remove much of the tedious overhead and re-work that comes with sifting through false positives. Contrast highlights an organization's true application attack surface by showcasing which third-party libraries are actually used during runtime, and validates vulnerability findings and fixes in custom code with real-time route intelligence.

By emulating the runtime behavior of the application, the Contrast Platform can determine what vulnerabilities can actually be exploited. This reduces false positives and noise, enabling teams to focus on vulnerabilities that can be exploited, instead of chasing alerts for potential threats. The platform applies this same methodology toward emerging cloud-native environments to monitor for vulnerabilities in serverless applications, with the ability to block threats in production.

The Bigger Truth

As software development continues to accelerate with the use of agile development practices and modern cloud infrastructure, all supporting tools and processes within the CI/CD pipeline must evolve to keep up.

- Vulnerable code is a code defect. Developers own code quality. It's time to give developers the control they need to be successful by fully embracing security as an attribute of building quality applications instead of a bolted-on, reactive, misaligned process that adds friction to agile development.
- It's time to free application security professionals from the soul-crushing, operational responsibility of triaging mountains of suspicious findings, enabling them to focus on improving application security programs, architecting effective tools strategies, and proactively coaching development teams.
- ESG recommends development teams take a step back to rethink application security strategies and consider new solutions from vendors like Contrast Security that are capable of more accurately identifying issues that matter while aligning with modern agile and cloud-driven development practices.

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