SANS

Analyst Program

Survey

SANS 2023 DevSecOps Survey

Written by **Ben Allen** and **Chris Edmundson**

August 2023

sonatype

Executive Summary

DevSecOps represents the intersection of software development (Dev), security (Sec), and operations (Ops) with the fundamental objective of automating, monitoring, and integrating security throughout all phases of the software development life cycle (SDLC): plan, develop, build, test, release, deliver, deploy, operate, and monitor. Ultimately, DevSecOps is fundamentally concerned with enabling agility, which inevitably brings with it the challenges that come with sharing the responsibility for security best practices with other stakeholders across the entire continuous integration/continuous deployment (CI/CD) pipeline. Achieving this shared-responsibility ideal requires the development of trusted relationships among development, security, and operations teams.

The 2023 DevSecOps Survey—the 10th in an annual series—considers a broad range of indicators of maturity in these areas and evaluates them against a retrospective view of previous years' survey responses, with the goal of helping security practitioners understand:

- How organizations use cloud platforms, architectures, and development ecosystems to identify security requirements, risks, and opportunities
- How organizations deploy appropriate security within the CI/CD pipeline, injecting security best practices while keeping up with the delivery of products and features to stakeholders
- What security tools and best practices organizations leverage to maintain a "shift-left" mentality—to keep security in mind continuously throughout the development process
- What skills are needed to empower organizations to architect secure applications and cloud services and help them find and fix vulnerabilities as early as possible
- What are the future trends and adoption rates of new technology, such as artificial intelligence (AI), data science, and GitOps—and how they might impact the future of DevSecOps

Key Findings

- **The trend toward organizations leveraging multiple cloud solutions continues, as indicated by the respondents using Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP) to run more than 75% of their application workloads. Forty-seven percent of the respondents said they use other cloud hosting providers, including Alibaba Cloud, IBM Cloud, and Oracle Cloud—a dramatic increase from just 25% last year.**
- **A key DevSecOps challenge remains the difficulty of acquiring the necessary funding to purchase newly available security and testing tools.**
- **The key success factors the survey respondents identified show the importance organizations continue to place on communications within the organization and creating "security champions" through professional development activities.**
- **Cloud-hosted virtual machines (VMs) are still preferred over containers and serverless functions, with 69% of respondents reporting at least 25% of their applications running on VMs.**
- **Another interesting trend this year is that DevSecOps is now clearly seen as a business-critical issue and a risk management concern. Forty percent of the respondents were aligned with the business side, and 13% of respondents identified themselves as business managers.**
- **The dominant industries represented by the respondents aligned with the technology, cybersecurity, and application development verticals. Representation from the banking and finance industry shows a significant decline (down from 17% in 2022 to 7% in 2023), and several key industries—notably government and healthcare—appear to be underrepresented, as they have been in past years.**
- **One of the notable forward-looking trends shown by the 2023 survey—reflecting industrywide trends—is a significant increase (16%) in respondents reporting that they were investigating and experimenting with the use of AI or data science to improve DevSecOps, up from 33% in 2022 to 49% in 2023.**

A Snapshot of the Respondents

The 363 respondents to this year's survey represent a highly diverse set of roles, industries, and organizational sizes (see Figure 1). Unsurprisingly, they show a strong bias toward security, with 34% of respondents performing a direct security function of some kind. Security administrator/security analyst is by far the most common role, at 10.2%. Development roles—including application developer, cloud architect, software engineer, and DevOps engineer—are, of course, also well represented, at 21%. But the single most represented role in the survey is business manager, at 13% of the respondents, clearly showing that DevSecOps is now broadly recognized as a business concern, not solely a technical issue. Management and executive roles, including the business manager role, comprise 40% of the respondents (including security and compliance managers, quality analysis [QA] release managers, CxOs, and IT managers/directors).

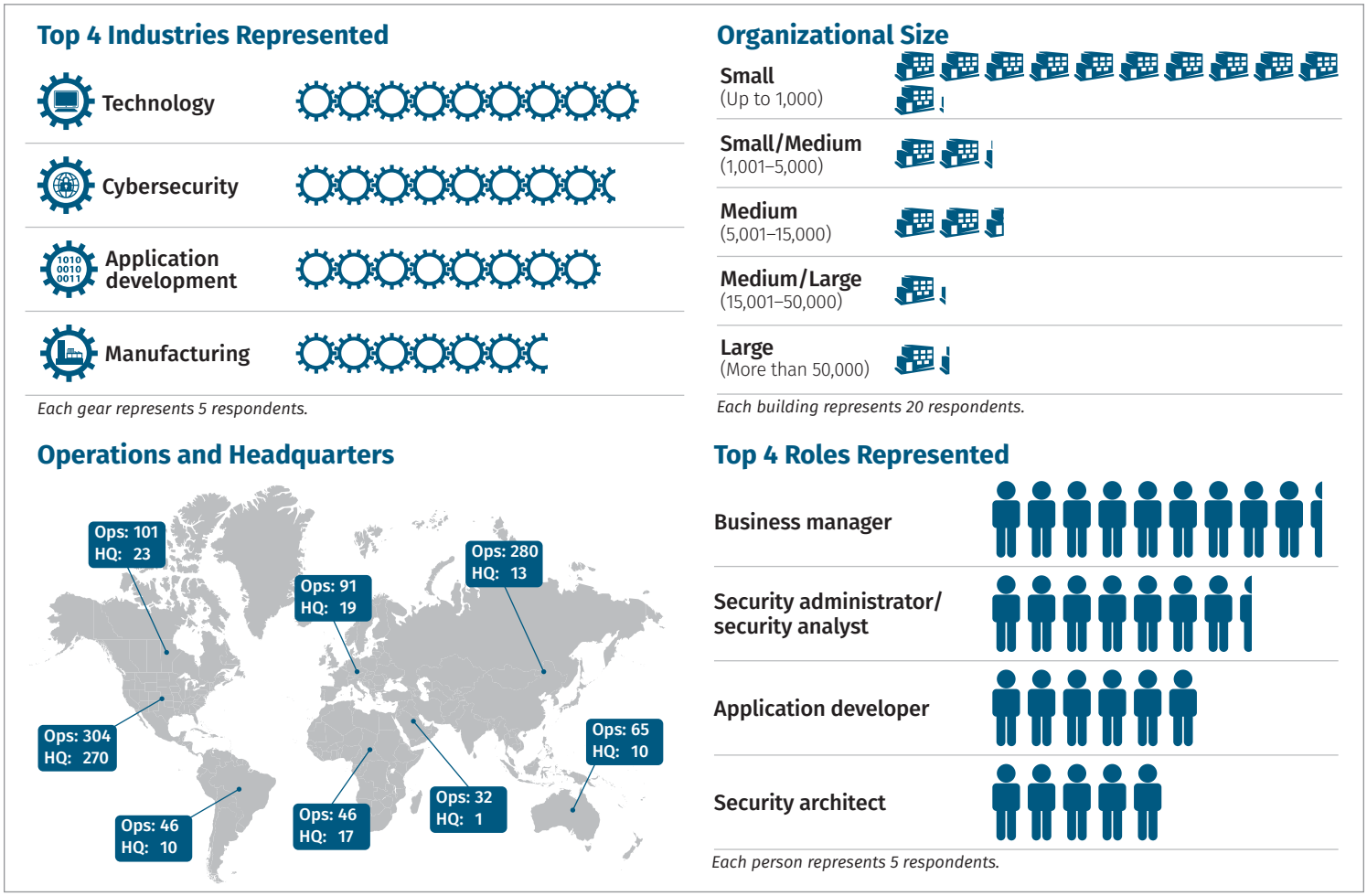

Figure 1. Demographics of Survey Respondents

More than half the respondents (53%) are from the top 5 industries. Small organizations—defined as those with 1,000 or fewer full-time and contract employees—dominate the survey, with a total of 61% of the respondents. Larger organizations are distributed relatively evenly across the other organizational-size categories. Additionally, this may also suggest that organizations are outsourcing their development resources.

The United States is disproportionately represented in terms of geography, with 74% of respondents' primary corporate headquarters located there and 84% of operations maintained there. Canada and Europe followed at a far-distant second and third, at 6% and 5% of corporate headquarters and 28% and 25% of operations, respectively.

Understanding the DevSecOps Environment

This year's survey, like the previous years', shows adoption and use of cloud computing as an IT delivery model continuing to accelerate dramatically. This year, for example, only 54% of respondents reported that their organizations run 25% or more of their applications on-premises, down from 65% in 2022 and 85% in 2021—a 31% drop in just three years (see Figure 2). Moreover, fewer than 5% of respondents indicated that they ran all their applications on-premises, while almost 7% said they have no on-premises applications at all, and more than 84% of the survey respondents reported at least some degree of cloud usage.

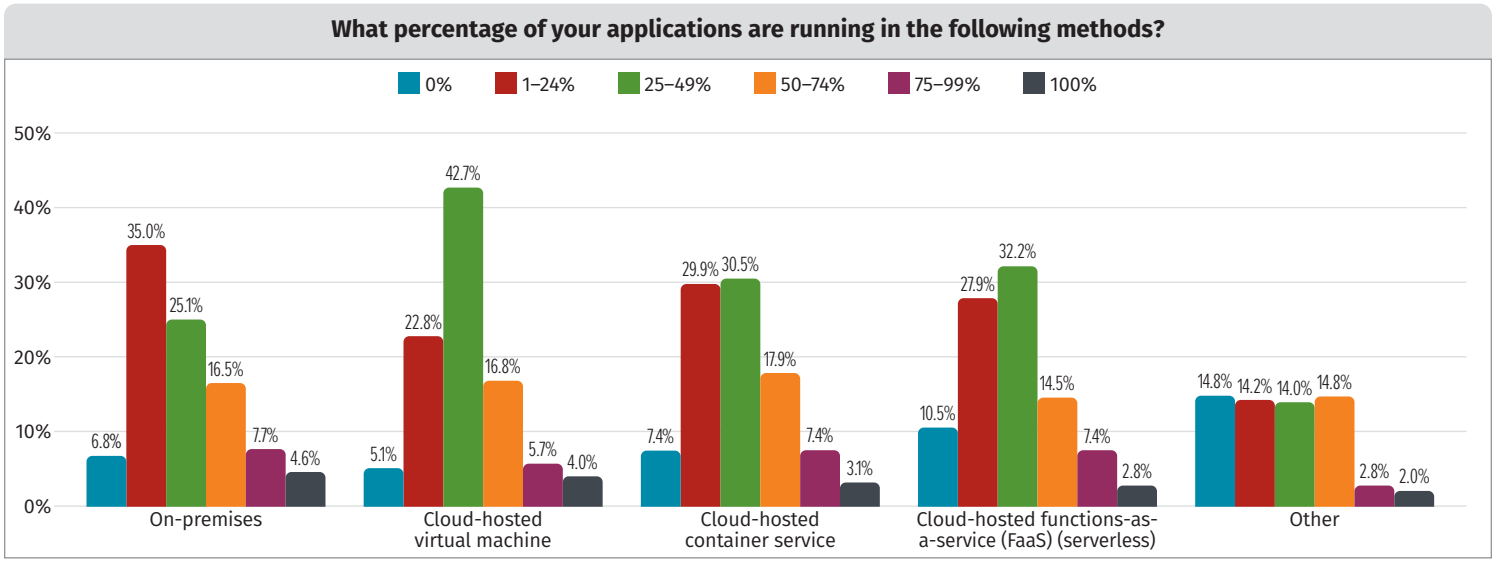

Figure 2. Most Commonly Used Platforms for Applications

These results clearly show that the shift to the cloud is ongoing (and almost certain to continue). But the survey results also offer an important reminder that not all applications are based in the cloud. Overall, a still very substantial 29% of the respondents reported that 50% or more of their applications remain on-premises.

A closer look at the cloud responses shows that, as in previous years, cloud-hosted VMs are still preferred over cloud-hosted container services or cloud-hosted functions-as-a-service (FaaS, also called serverless computing)—but also that most organizations' cloud implementations remain highly diverse. In this year's survey, although 69% of the respondents said that at least 25% of their applications run on VMs, 59% use cloud-hosted container services for the same percentage of their applications and 57% use FaaS.

This mix of VMs, containers, and FaaS has important security implications, because all three of these technologies must be properly secured. That means DevSecOps teams must have the skills and tooling to secure all three approaches—which, despite considerable overlap, are all distinctly different. A further consideration is that an enterprise using the cloud can likely delegate some mundane security tasks to its cloud service provider—freeing up its own personnel for more important higher-level duties—but this is not the case with on-premises applications. This suggests that organizations using the cloud should look to providers that are prepared to take on more security management responsibilities.

TAKEAWAY

DevSecOps teams need to invest in tools that make it possible to secure their workloads effectively, wherever they run. Software composition analysis (SCA), static application security testing (SAST), dynamic application security testing (DAST), and threat modeling tools, for example, can all be used to improve the security of long-lived VMs and short-lived containers or functions. When selecting tools, security practitioners must recognize that different runtime environments need different tools and must consider whether cloud providers—both current and prospective—have tools integrated into their ecosystems that could potentially streamline security workflows.

Application Hosting in the Cloud

The survey results clearly show that most organizations using the cloud are engaging with multiple cloud service providers and that the distribution of applications running on each of the three most important cloud service providers is beginning to even out.

For the more than 84% of respondents who reported using the cloud:

- 90% have applications running on Amazon Web Services (AWS).
- 84% have applications running on Microsoft Azure.
- 74% have applications running on Google Cloud Platform (GCP).

Moreover, 47% said they use other cloud hosting providers, including Alibaba Cloud, IBM Cloud, and Oracle Cloud—a dramatic increase from just 25% last year.

Another important finding is the clear trend away from using a single cloud hosting provider to run the majority of an organization's workloads. Table 1 shows the percentage of respondents to the 2021, 2022, and 2023 surveys who reported using AWS, Azure, or GCP to run 75% or more of their applications. Figure 3 details the complete distribution of the 2023 survey responses.

Table 1.
Respondents Concentrating 75% or More of Workloads on a Single CSP, 2021–2023

	AWS	Azure	GCP
2023	17.3%	11.2%	9.8%
2022	21.6%	14.8%	5.9%
2021	27.1%	18.4%	7.2%

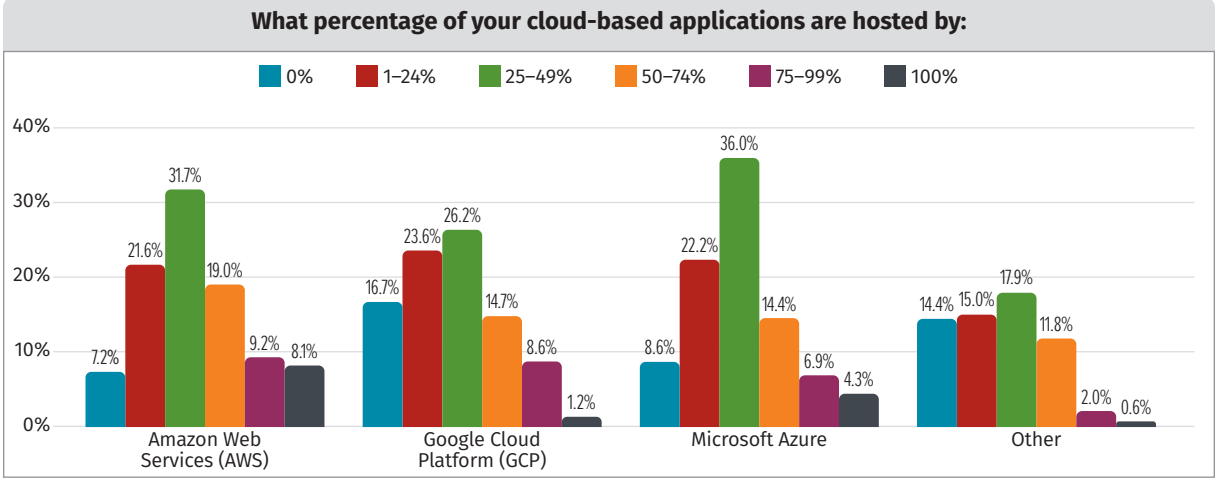

Figure 3. Extent of Cloud-Based Application Hosting

There are many reasons, including the need for business continuity planning and the desire for negotiation leverage, that an organization may choose to distribute its workloads across multiple cloud service providers. The benefits of using multiple cloud service providers are obvious, but so are their security implications: Each provider's environment must be properly secured, but every environment works differently and presents different security challenges. And the work involved increases exponentially with each additional provider used.

One way leading DevSecOps teams are coping with the multicloud challenge is by creating platform-agnostic applications, typically using containerization, so that the application can run in any cloud service provider's container service, or even on-premises, with the necessary infrastructure in place.

Securing Multicloud Environments

The increasing reliance on multiple cloud service providers, with a mix of VMs, containers, and FaaS, also sharply increases the challenges of ensuring that all those cloud resources are properly secured. To evaluate this challenge, the 2023 DevSecOps survey considered similar results from identical questions asked in the 2022 and 2023 surveys regarding two of the most important sets of cloud security tools:

- To what extent has your organization adopted cloud security posture management (CSPM) software, either commercial or open source? (See Figure 4 for years 2022/2023.)
- To what extent has your organization adopted cloud workload protection platform (CWPP) software? (See Figure 5 for years 2022/2023.)

The survey results from these two years show that even though CSPM is widely deployed, it is still highly underutilized. Most respondents said they are using either a commercial or an open source CSPM tool, but fewer than 16% overall (2023) and 21% overall (2022) said they use those solutions for 75% or more of their AWS accounts, Azure subscriptions, or GCP projects.

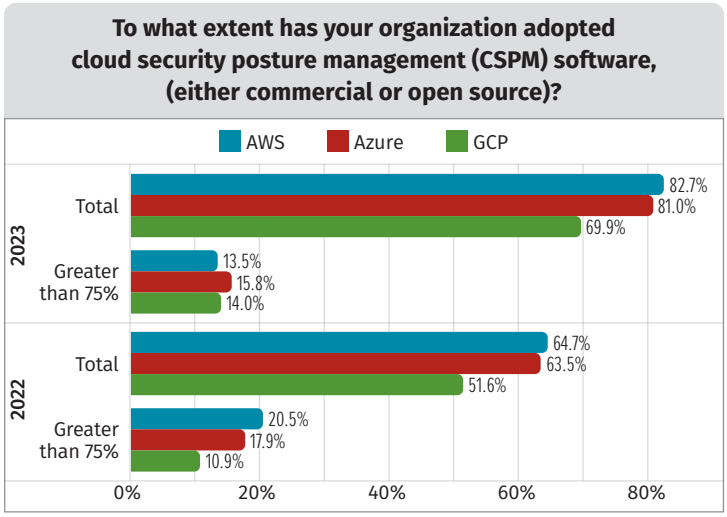

Figure 4. Extent of CSPM Adoption

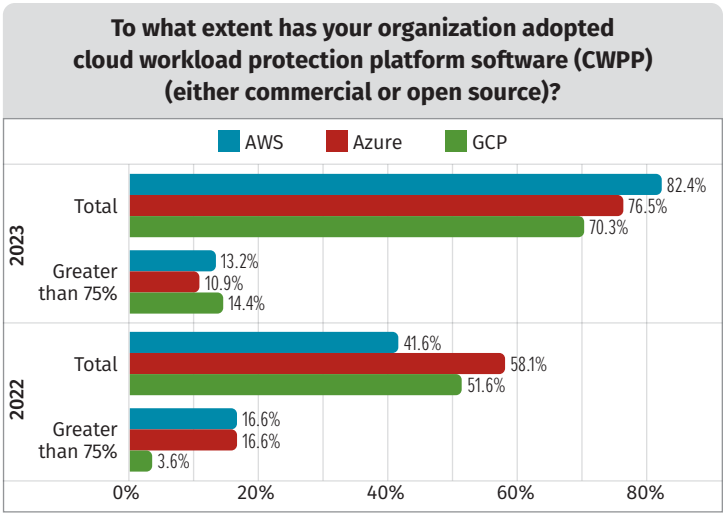

Figure 5. Extent of CWPP Adoption

This decline from the previous year could have several causes. One cause might be that the increase in the use of these tools has made organizations more aware of their cloud provider inventories and the weaknesses in their protections. Another cause might be that vendor pricing is driving organizations to make difficult choices between inadequate protection and unacceptable cost.

CWPP products are also very much underutilized. Although a majority of the respondents (greater than 70% across all platforms) said their organizations use CWPP, a much smaller percentage (less than 15% overall in 2023, down slightly from 17% in 2022) use it in at least 75% or more of their AWS accounts, Azure subscriptions, or GCP projects.

Both findings suggest that DevSecOps teams are missing a valuable opportunity to enhance the security of their cloud environments. CSPM software can help DevSecOps teams ensure that the cloud environments that host their applications are properly configured and secured using industry best practices, but only if this software is used consistently across all cloud accounts. Similarly, CWPPs provide various security services for workloads, regardless of whether the work is performed by VMs, containers, or serverless computing. In the past, this would typically have required the installation of multiple agents, resulting in a drain on VM resources, but CWPP solutions have evolved to overcome that problem.

TAKEAWAY

Both CSPM and CWPP are essential capabilities for organizations operating in multicloud environments. As an organization moves further away from reliance on a single cloud hosting provider, the work of securing each cloud environment increases exponentially. Organizations should consider using or increasing their adoption of commercial or open source CSPM software to ensure that each cloud environment is securely configured, and they should implement CWPPs to protect application workloads at execution time.

Security at Velocity

The survey results clearly show that delivery velocity—the speed at which changes are made to applications in development—is remarkably stable. When asked how often their organizations deliver system changes to production, 54% of respondents said at least weekly, with 24% reporting that changes are delivered at least once per day or on a continuous basis. The distribution of delivery times has been fairly consistent for the past three years, with a slight dip this year in the "daily" and "continuous" categories (see Figure 6).

On average, how often do you deploy system changes to production applications? ***Select the most appropriate answer.***

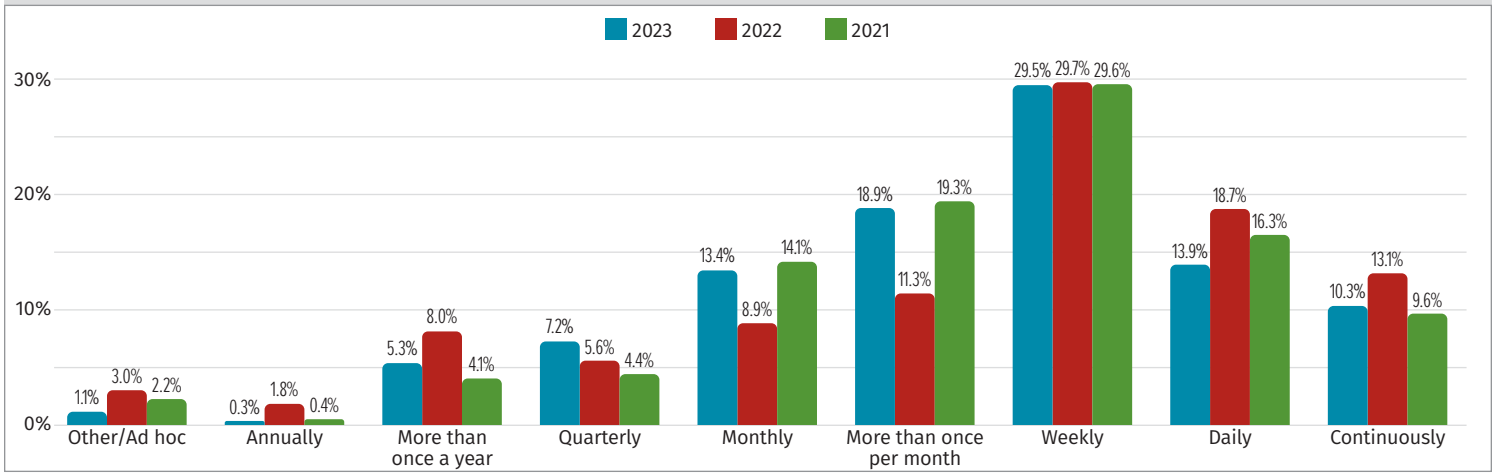

Figure 6. Frequency of Delivery to Production, 2021–2023

Investments in continuous integration/continuous deployment (CI/CD) tooling enable organizations to make small changes to their production codebase faster, with many teams working to deliver a constant stream of changes that can be pushed to production. The high ratio of developers to security engineers makes it clear that the only way to keep pace is to automate security testing in the CI/CD pipeline so that every code push is evaluated for security flaws.

The fact that approximately 45% of respondents reported deploying changes on a weekly or daily basis, but not continuously, suggests that DevSecOps teams may now have more time available to run deeper scans, between code commit and release to production, while also meeting business delivery requirements.

Organizational leaders should look for meaningful metrics that make it possible to ensure that 100% of the application portfolio is deployed using CI/CD pipelines complete with security tests. Once all applications have integrated security testing performed at every pass through the pipeline, new security tests can be introduced to raise the bar until all security requirements are achieved. It's important to remember that security tests can only be designed to test for *known* issues. For this reason, penetration tests and bug bounties—which can help security practitioners find *unknown* issues—still have an essential role to play in a comprehensive application security program. Cloud-native application protection platforms are being used to characterize normal application behavior and enforce zero-trust principles as an additional countermeasure to protect against exploited security flaws (see the "DevSecOps Tools and Practices: What Works?" section for more details).

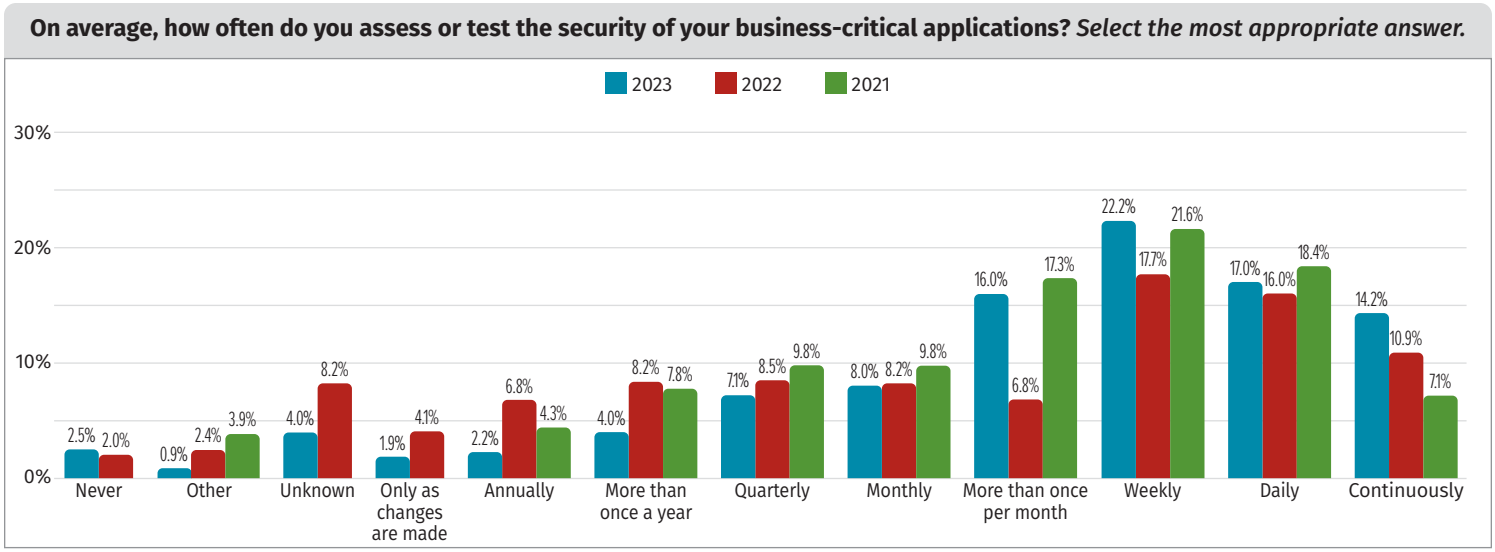

Figure 7. Frequency of Assessing Business-Critical Applications, 2021–2023

New security threats, flaws, and vulnerabilities are, of course, being discovered constantly. Even an application with a stable codebase can have security flaws that remain undiscovered until the application is subjected to security testing. When most organizations (54%) are delivering application changes to production at least weekly, the only way to cope with this volume of activity is to automate security testing and integrate it with the CI/CD tool chain.

The implementation of automated security testing requires significant investment. But once this investment has been made, organizations can utilize these capabilities to incrementally improve the security of the applications they build, write custom tests to assess all their applications, and quickly assess the impact of newly discovered vulnerabilities across their application portfolio.

To explore this trend, we asked, "On average, how often do you assess or test the security of your business-critical applications?" (See Figure 7 above.) The responses were striking:

- 53% of respondents said their organizations test the security of their business-critical applications at least weekly, with 31% testing them at least daily.
- Comparing the share of organizations that are *deploying* applications weekly or more frequently (54%) with that of organizations *testing* their business-critical applications at least weekly (53%) indicates that integrated automated security testing with DevOps tooling is becoming the norm.

TAKEAWAY

Security organizations should automate as much of the security testing process as possible, so that testing can be performed more frequently, more broadly, and more cost-effectively.

Security at velocity also involves remediation speed, of course. Automated security testing is highly effective at identifying known security vulnerabilities, but remediating critical security issues takes engineering time and management commitment. The 2023 distribution of responses looks much like the responses from 2022, with 53% and 54% of respondents, respectively, stating that their organizations get critical security issues resolved within a week or less.

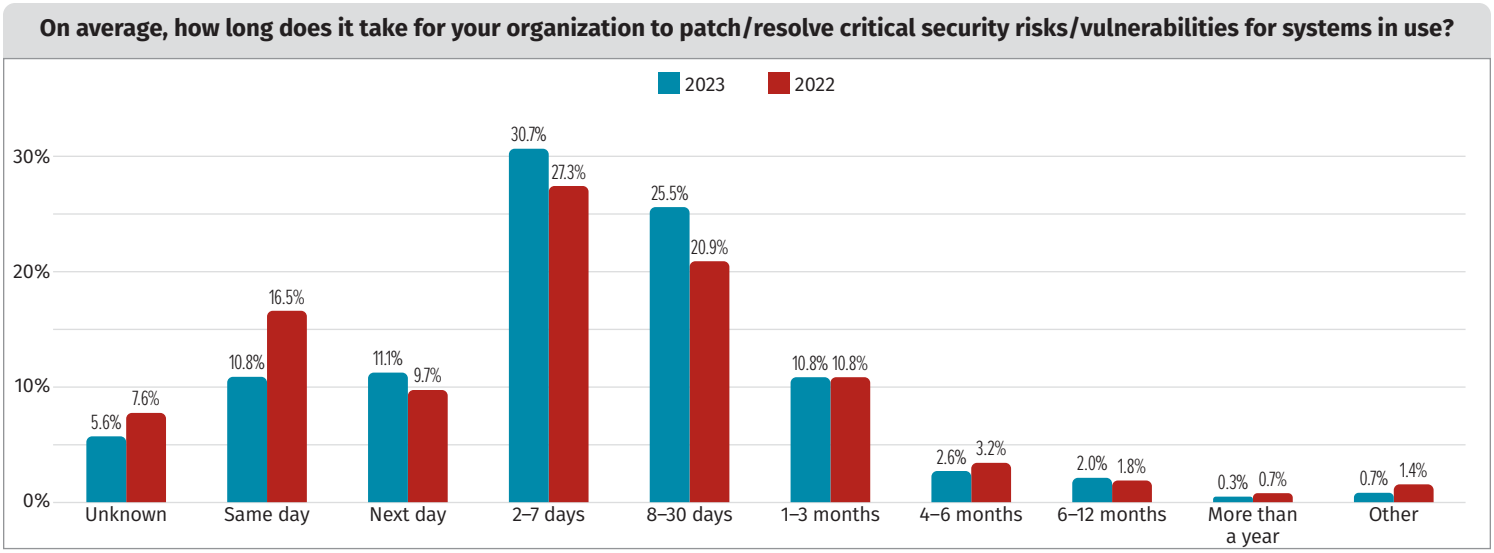

Figure 8. Average Time to Resolve Vulnerabilities, 2022–2023

The tail of the distribution, however, shows that 16% of the respondents are aware that it takes their organization more than 30 days to fix critical security issues. Development teams often feel pressure from management to prioritize new functionality over the maintenance of application security. And whereas creating new functionality is interesting, most people consider patching security issues to be drudgery. To help development teams address issues in components included in their applications, numerous SCA tools (for example, Mend SCA, Snyk Open Source, Synopsys Black Duck,[1] and Veracode SCA) include the ability to integrate with source code management systems to create a pull request that developers can review, test, and merge into the feature code as part of their workflows, reducing some of the burden on them and also reducing the time to repair the vulnerability (see Figure 8).

Automated Compliance

Policy-as-code and CSPM are different techniques for enforcing compliance policies automatically. In this year's survey—like the previous year's—more than 60% of respondents (62% in 2023, 60% in 2022) said that at least 50% of their organization's compliance enforcement is automated. Still, the number of respondents who reported that 100% of their policies are enforced automatically decreased significantly in 2023 from 2022 (6% versus 18%). There was also a decrease in the percentage of respondents who either hadn't implemented any automated policy enforcement or didn't know how much coverage their automated policy checks had. (These responses show as negative percentages in Figure 9.)

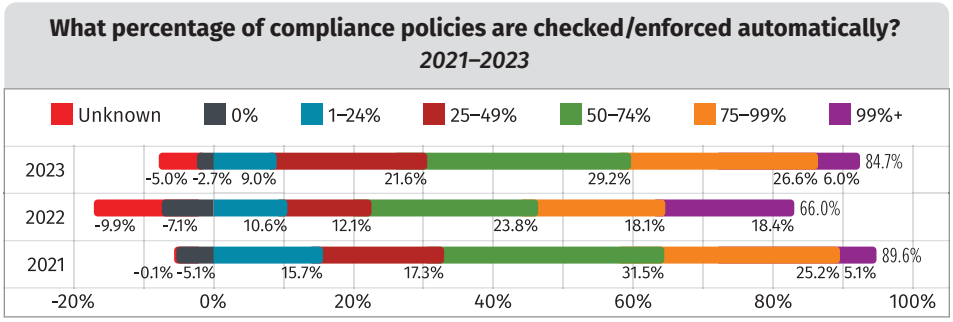

Figure 9. Percentage of Compliance Policies Checked or Enforced Automatically

The use of automated checking or enforcement of compliance policies shows that DevSecOps principles are starting to have a more significant impact on security practices. Security teams have begun to recognize the importance of implementing DevOps principles within their own practices to achieve enterprise scale and development agility. At the same time, DevOps teams are integrating policy-as-code tests into their CI/CD pipelines to validate security policy compliance. These tests have value that extends beyond compliance, because they're cost-effective: Writing a security test has a one-time cost that quickly approaches zero per test when that test is performed frequently. Both practices are helping organizations meet the goal of scalable continuous compliance.

Securing Container Services

We've already seen that as organizations move to the cloud, they deploy their applications using a combination of VMs, FaaS, and containerized workloads. Whereas VMs offer a strongly self-supported model that corresponds to on-premises data center environments, and FaaS offers a mostly cloud-provider-supported model, the containerized workloads space includes a wide range of support models between the other two, so it's worthwhile to take a closer look at how container services are being used.

Organizations looking to use container orchestration tools face three basic questions. The first question is whether to use Kubernetes, Docker Swarm, or some other orchestration option. The second question is whether to use such a tool as a managed service or to manage it themselves, and the third question is whether to run on cloud-hosted or on-premises infrastructure. Figure 10 shows the choices of container orchestration tools that respondents' organizations have made over the past three years of the survey:

- Cloud hosted is more prevalent than cloud managed for both container services and Kubernetes.
- For on-premises organizations, the choice between an orchestrator (Kubernetes or OpenShift) and a container engine (Docker or Docker Swarm) is an even split for 2023.
- Cloud-managed container services had an approximately 10% increase this year, suggesting that as organizations migrate to the cloud and to containers, they're favoring cloud-managed container services over cloud-managed Kubernetes services.

Which container orchestration tools are managing your production workloads? ***Select all that apply.***

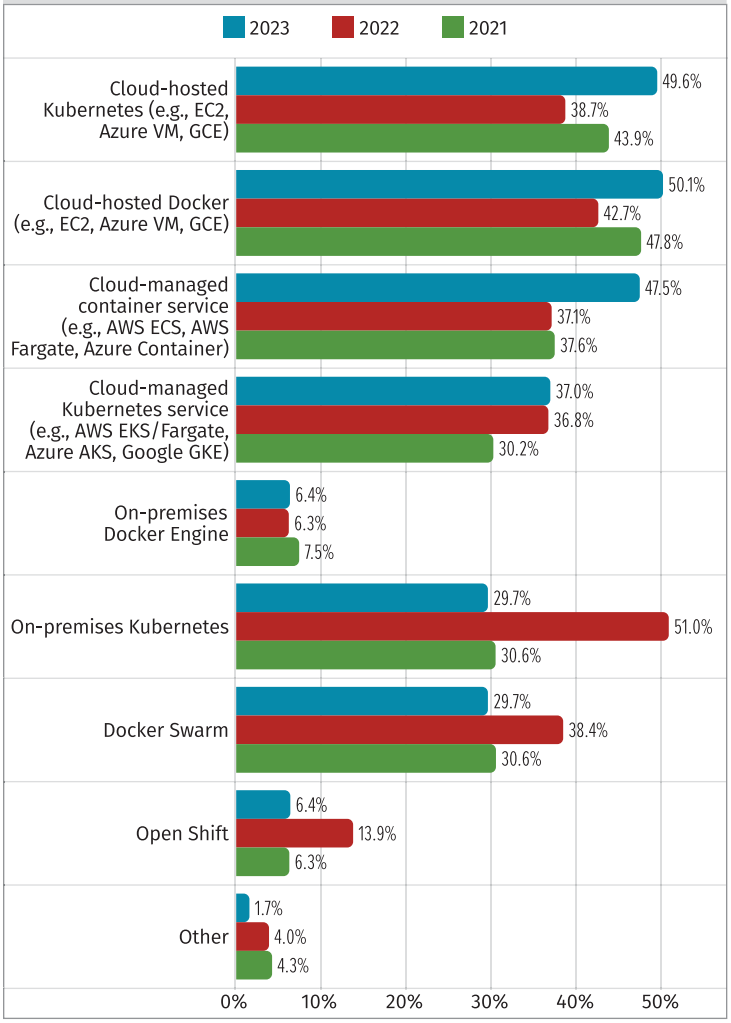

Figure 10. Container Orchestration Tools Used to Manage Production Workloads

TAKEAWAY

When moving containerized workloads to the cloud, many organizations seem to be taking a lift-and-shift approach, moving their on-premises VMs, which host Kubernetes or Docker environments, to cloud-hosted VMs that perform the same functions. As the provider-managed offerings for Docker and Kubernetes mature, this "lift-and-shift" approach may create challenges for organizations trying to achieve deeper integration with their cloud providers' security tools. That lift-and-shift approach may, however, also reflect an intentional choice by organizations to avoid these deep integrations as part of their multicloud strategy.

Programming Environments and Risks

Surprisingly, the top 4 responses to the question of which languages and platforms present the greatest risk to their application portfolios show no overlap at all with 2022's answers. (The 2023 survey shows Python as the greatest risk by a wide margin—at least 12% greater than the next option, C/C++.) Even so, the responses concerning the top 10 language/platform risks show broad stability over the past three years of the survey (see Figure 11), despite significant fluctuations in the respondents' demographics during that time period.

Whichever language or languages are seen as the riskiest, the most popular, or the most intriguing at any given time, organizations need to develop processes and establish standards for bringing new languages into their portfolios. These initiatives should consider factors like memory safety, support for CI/CD tools (including linting, coding standards, security scanning, and dependency management), and the need to ensure interoperability with languages already in use when defining organizational rules for adopting a new language. It is also extremely important that organizations consider the most dangerous known software errors when formulating standards for adopting new languages in the enterprise. (The regularly updated CWE/SANS TOP 25 Most Dangerous Software Errors[3] list is an excellent source for this information.) Understanding the dangers these errors present—and determining which capabilities are required to identify, remediate, and prevent them—will enable informed decision-making that improves the organization's overall security posture.

Languages with strong memory safety features integral to their design—for example, Go, Rust, Scala, and Swift—continue to be perceived as comparatively low risk.[3] The recommendation to migrate to memory-safe languages for new projects presented in the 2022 DevSecOps Survey is now widely supported, as evidenced by publications from sources as wide-ranging as the National Security Agency (NSA)[4] and *Consumer Reports*.[5]

TAKEAWAY

Identifying the most dangerous software errors and how they can be eliminated will be critical to the success of DevSecOps teams' ongoing efforts to reduce bugs and deliver more stable systems. The adoption of memory-safe languages, particularly on new projects, can eliminate entire classes of vulnerabilities.

Which languages and platforms in your application portfolio have been the greatest source of risk or exposure to your organization? *Select your top three.*

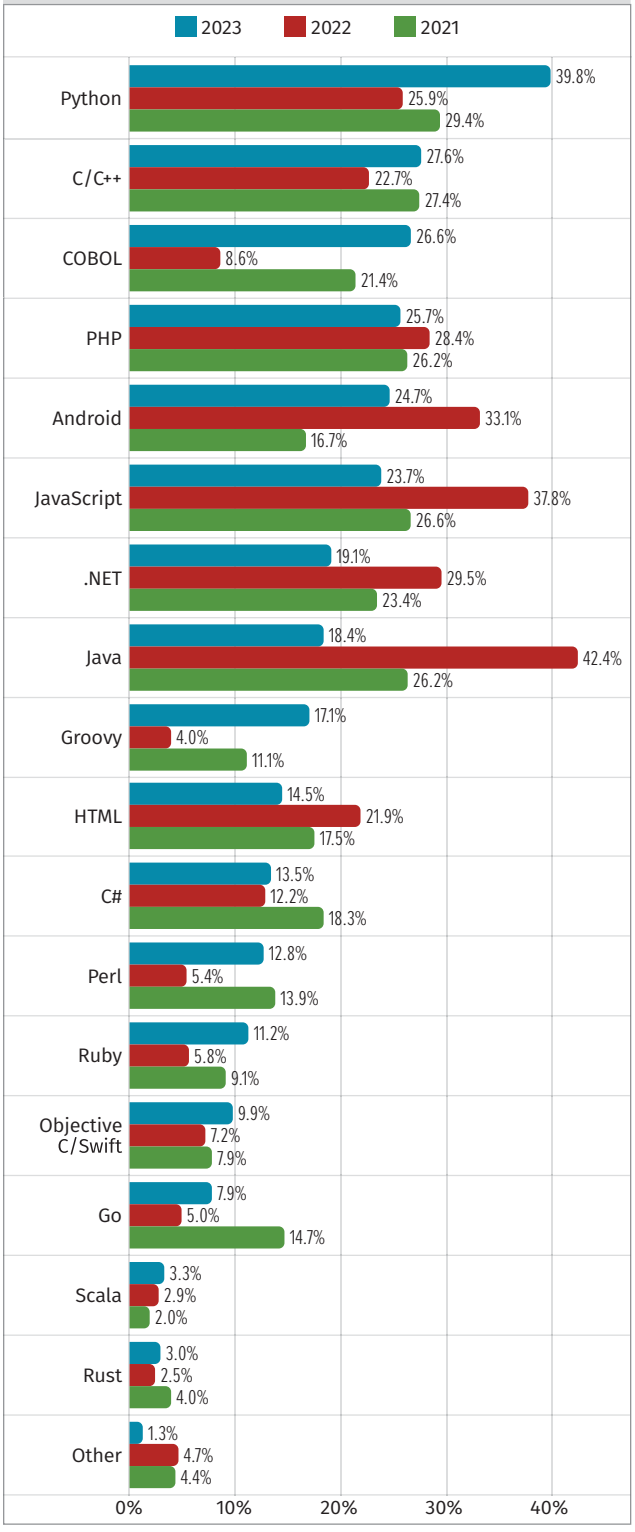

Figure 11. Languages and Platforms Presenting the Greatest Risk or Exposure, 2021-2023

[2] "CWE/SANS TOP 25 Most Dangerous Software Errors," www.sans.org/top25-software-errors/

[3] "What Is Memory Safety and Why Does It Matter?," www.memorysafety.org/docs/memory-safety/

[4] "Software Memory Safety," www.nsa.gov/Press-Room/News-Highlights/Article/Article/3215760/nsa-releases-guidance-on-how-to-protect-against-software-memory-safety-issues

[5] "Report: Future of Memory Safety," https://advocacy.consumerreports.org/research/report-future-of-memory-safety/

DevOps Foundational Practices

The role of the cloud security architect in supporting DevSecOps process improvements shows a small decrease compared with last year (2%). This is, however, offset by an overall increase in organizational focus on process improvement—from 79% in 2002 to 84% in 2023. There is also a shift in DevSecOps process improvement focus to a more distributed effort—with a decline in cloud security architects' focus on DevSecOps process improvement offset by an increase in focus spread across other teams. Whether DevSecOps process improvement should be driven by a cloud security architecture team or distributed across other development, operations, and security teams will vary from organization to organization; the decision should ultimately be driven by the need to align with the organization's underlying values and structure. Wherever an organization's DevSecOps process improvement efforts are focused, the overall growth in active efforts in this area shown in the survey is a sign that organizations are getting a good return on their DevSecOps investments (see Figure 12).

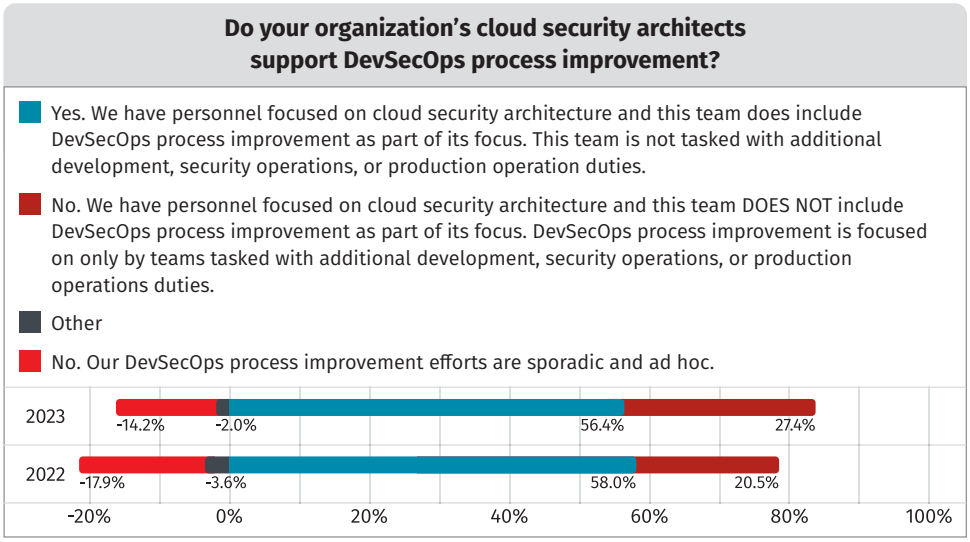

Figure 12. Role of Cloud Security Architects Supporting Process Improvement

This year's survey shows a strong preference for purpose-built commercial CI tools for build and release automation—a reversal of 2022's dramatic shift toward on-premises open source tools. As suggested in our analysis of the 2022 survey, the mix of preferences for CI tools likely has linkages to other properties of the respondent pool: Whether workloads run on-premises or in the cloud, compliance requirements and the size and budget of the organization will all shape an organization's selection of CI tools (see Figure 13).

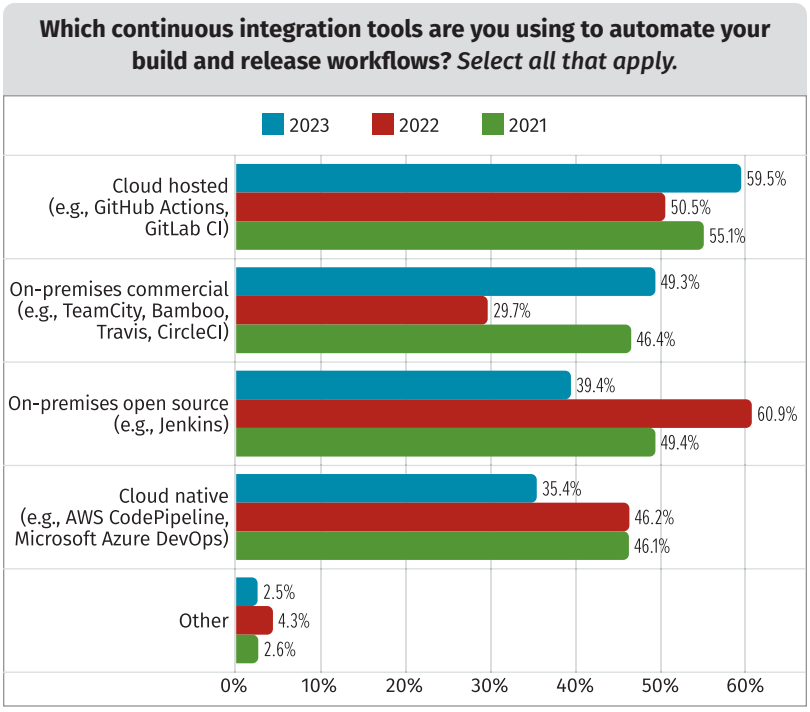

Figure 13. Continuous Integration Tools Usage, 2021–2023

Security Testing

The 2023 survey shows the highest percentage of testing (49%) being conducted at the code commit/pull request stage. Across the spectrum of options, security testing is down overall, with a marginal increase in security unit testing. The emphasis on security testing prior to coding—architecture/design and requirements/use cases—that was seen in 2022 has declined sharply this year, and testing while coding via integrated development environment (IDE) plug-ins has declined, as well. The "shift security left" mentality seems to be less pervasive among this year's respondents, which could be attributed to the shift in the industries and roles represented (see Figure 14).

The changes from last year to this in both top roles and top industries represented suggest that organizations in highly regulated industries (notably banking and finance) prioritize security testing to meet their regulatory requirements. Additionally, in nearly every category, fewer respondents indicated that they perform security testing in each phase of the build and release pipeline workflow.

When asked about the tools, practices, or techniques their organizations use, the 2023 survey respondents identified "upfront risk assessments before development starts" as the most useful item. Given the high value of upfront risk assessments, it seems unfortunate that 9% fewer survey participants than last year are performing security testing during those risk assessments as part of their workflows.

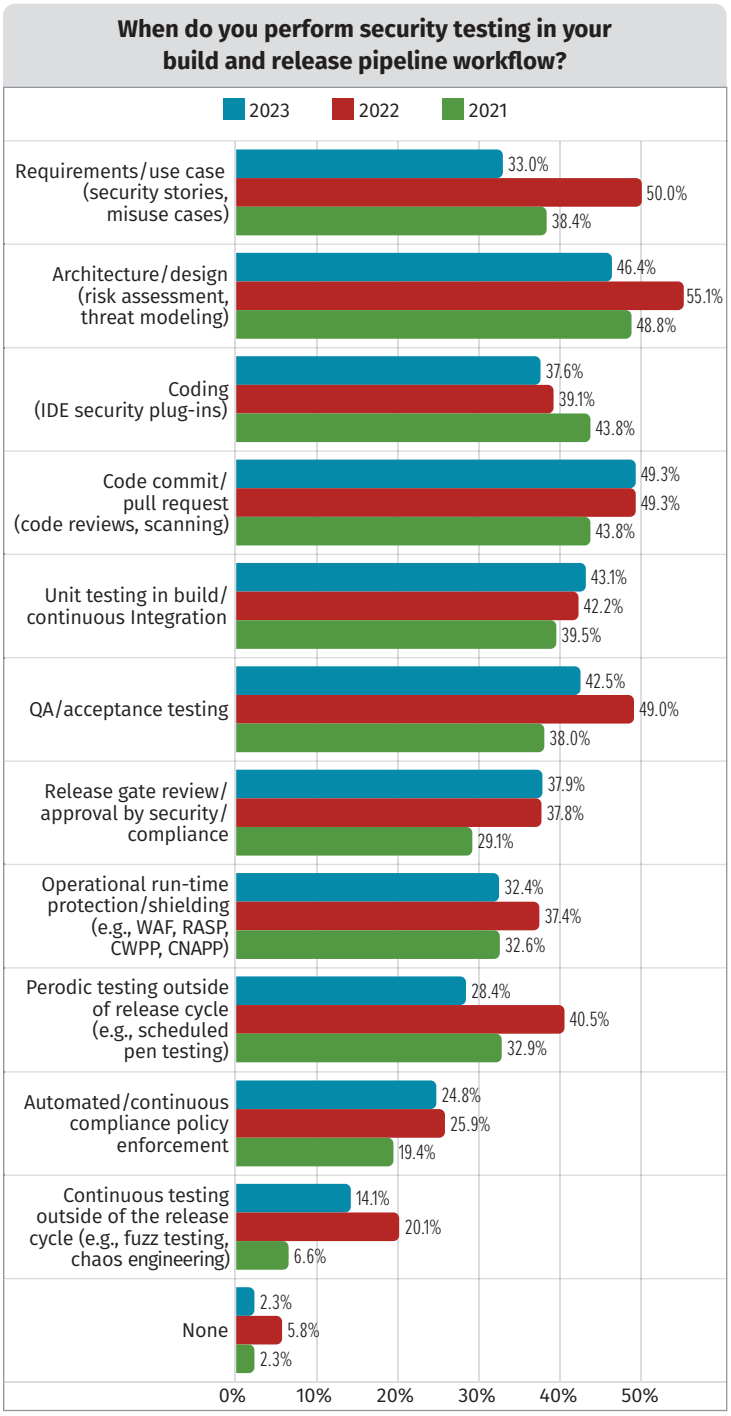

Figure 14. The Timing of Security Testing in Build and Release Pipeline Workflows, 2021–2023

Manual code review continues to be widely perceived as not useful, despite moving up in rank from 14th to 10th most useful in the past year. Nonetheless, over 95% of respondents reported using manual code review, despite their evident distaste for it. This polarization surrounding the usefulness of manual code review becomes especially concerning when coupled with pull request/code commit being the most popular time to perform security testing, because if manual code review is not valued, the likelihood of sloppy or rushed reviews resulting in security flaws increases (see Figure 15).

What application security tools, practices, or techniques do you use in your organization?
Rate those processes in terms of their usefulness. Indicate N/A for those that you don not use.

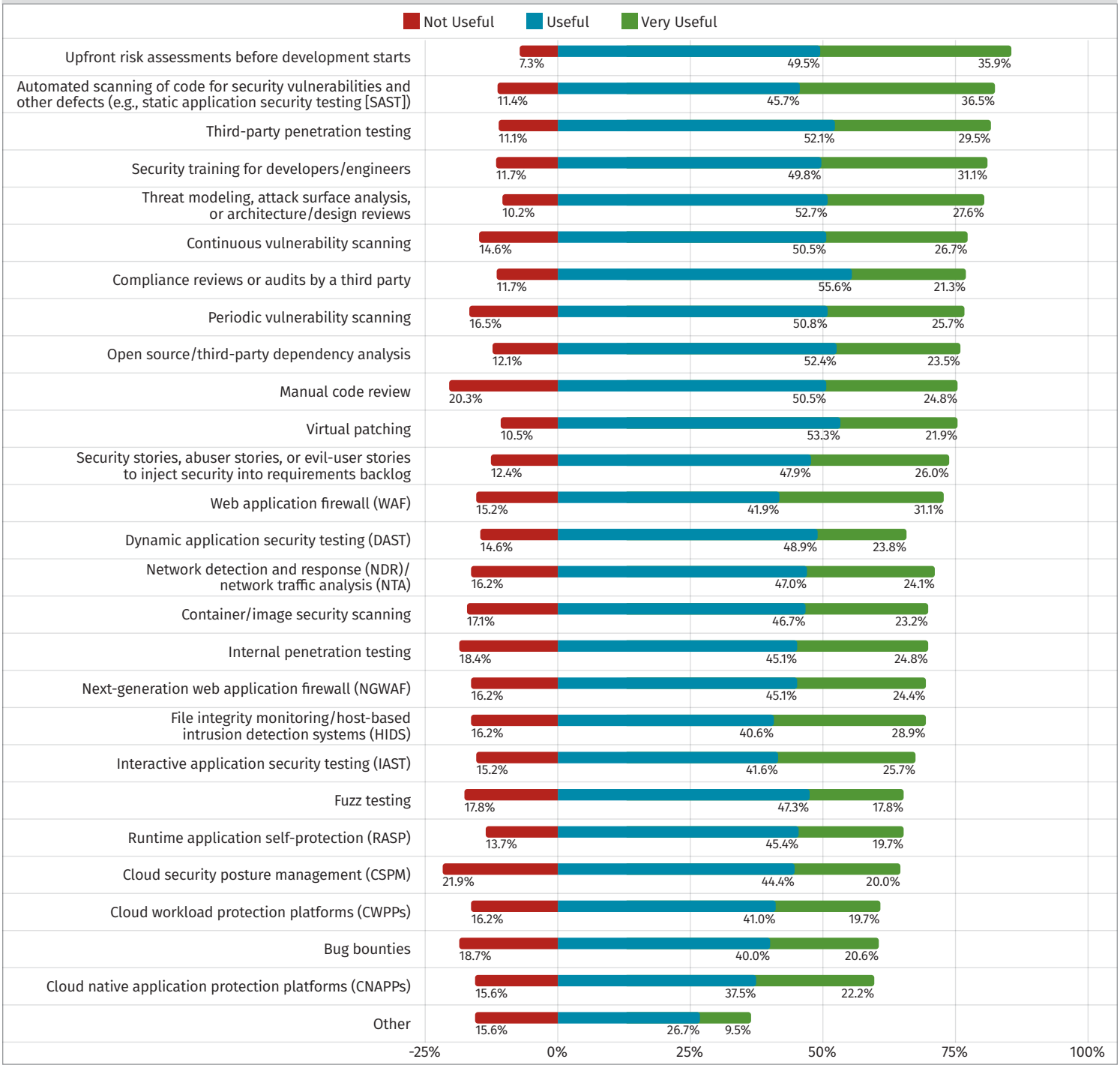

Figure 15. Usefulness of Various Security Testing Practices and Tools

Table 2. 2023 Survey Results on Usefulness of Various Security Testing Practices and Tools

	Not Useful	Useful	Very Useful	Total Useful	Change	Rank 22➔23
Upfront risk assessments before development starts	7.3%	49.5%	35.9%	85.4%	+8	9➔1
Automated scanning of code for security vulnerabilities and other defects (e.g., static application security testing [SAST])	11.4%	45.7%	36.5%	82.2%	+2	4➔2
Third-party penetration testing	11.1%	52.1%	29.5%	81.6%	+4	7➔3
Security training for developers/engineers	11.7%	49.8%	31.1%	80.9%	-1	3➔4
Threat modeling, attack surface analysis, or architecture/design reviews	10.2%	52.7%	27.6%	80.3%	+8	11➔5
Continuous vulnerability scanning	14.6%	50.5%	26.7%	77.2%	-1	5➔6
Compliance reviews or audits by a third party	11.7%	55.6%	21.3%	76.9%	+10	17➔7
Periodic vulnerability scanning	16.5%	50.8%	25.7%	76.5%	-6	2➔8
Open source/third-party dependency analysis	12.1%	52.4%	23.5%	75.9%	+1	8➔9
Manual code review	20.3%	50.5%	24.8%	75.3%	+4	14➔10
Virtual patching	10.5%	53.3%	21.9%	75.2%	+8	19➔11
Security stories, abuser stories, or evil-user stories to inject security into requirements backlog	12.4%	47.9%	26.0%	73.9%	+4	16➔12
Web application firewall (WAF)	15.2%	41.9%	31.1%	73.0%	-12	1➔13
Dynamic application security testing (DAST)	14.6%	48.9%	23.8%	72.7%	+1	15➔14
Network detection and response (NDR)/network traffic analysis (NTA)	16.2%	47.0%	24.1%	71.1%	-5	10➔15
Container/image security scanning	17.1%	46.7%	23.2%	69.9%	-4	12➔16
Internal penetration testing	18.4%	45.1%	24.8%	69.9%	-11	6➔17
Next-generation web application firewall (NGWAF)	16.2%	45.1%	24.4%	69.5%	-5	13➔18
File integrity monitoring/HIDS	16.2%	40.6%	28.9%	69.5%	-1	18➔19
Interactive application security testing (IAST)	15.2%	41.6%	25.7%	67.3%	+1	21➔20
Fuzz testing	17.8%	47.3%	17.8%	65.1%	+3	24➔21
Runtime application self-protection (RASP)	13.7%	45.4%	19.7%	65.1%	+3	25➔22
Cloud security posture management (CSPM)	21.9%	44.4%	20.0%	64.4%	-3	20➔23
Cloud workload protection platforms (CWPP)	16.2%	41.0%	19.7%	60.7%	-2	22➔24
Bug bounties	18.7%	40.0%	20.6%	60.6%	+1	26➔25
Cloud native application protection platforms (CNAPP)	15.6%	37.5%	22.2%	59.7%	-3	23➔26
Other	15.6%	26.7%	9.5%	36.2%	+0	27➔27

Something to watch for in next year's DevSecOps survey—given the sudden and dramatic emergence of AI technologies—will be how AI coding-assist tools impact the shape of these practices, and whether they change perceptions of the value of manual code review.

This year's survey includes some dramatic changes in how respondents valued selected application security tools, techniques, and practices. Table 2 shows the perception of usefulness for the application security tools, practices, and techniques rated by survey respondents, with the most useful listed first. The Change column shows changes to the ranking order from 2022 to 2023. Some noteworthy changes include:

- Third-party compliance reviews or audits moved up 10 places, despite the large decline in demographics of survey respondents from banking and finance industries.
- "Threat modeling, attack surface analysis, or architecture/design" and "upfront risk assessments before development starts" both moved up eight positions, with the latter seen as most useful overall this year. Upward movement in these two categories epitomizes shifting security left, toward work that can be done before a single line of code is even written.

- The 2023 respondents viewed a web application firewall (WAF) and internal penetration testing options as much less useful than 2022's cohort (drops of 12 and 11 positions, respectively). This reinforces the perception among those surveyed that early intervention is critical to success.

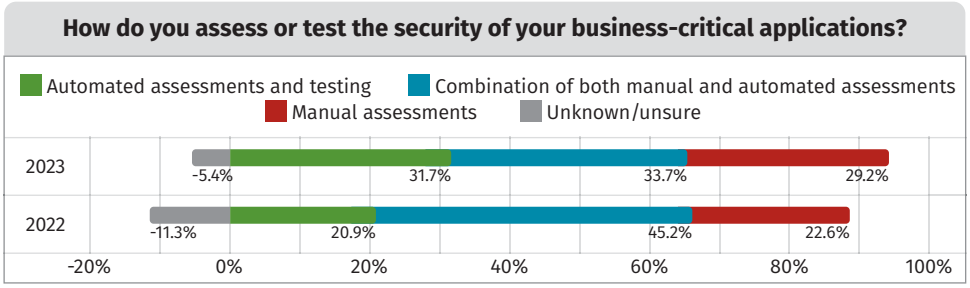

Figure 16. Automated/Manual Assessment of Business-Critical Applications

The survey question asking how respondents assess or test the security of business-critical applications shows an increase in understanding of how testing is performed compared with the 2022 survey results, and a significant portion of that increase is focused on automated testing. The increase in both purely automated and purely manual testing is offset by a reduction in hybrid testing (see Figure 16). To get an idea of why that shift occurred, let's take a look at who performs the security testing.

When asked who performs security testing for organizations, responses indicated a decrease in testing being performed by the internal security team and increases in testing by both external consultants and cloud-based testing platforms. These sets of changes together can be explained by viewing the external consultants as "purely manual" testing, external cloud-based security testing platforms as "purely automated" testing, and internal teams as a combination of the two (see Figure 17).

This year's survey shows a shift in security testing from internal staff to external partners and vendors. However, because the ratios in 2023 and 2021 resemble each other in a manner similar to the demographics of survey respondents, this may be more a reflection of the makeup of the survey participants than an indication of industry trends.

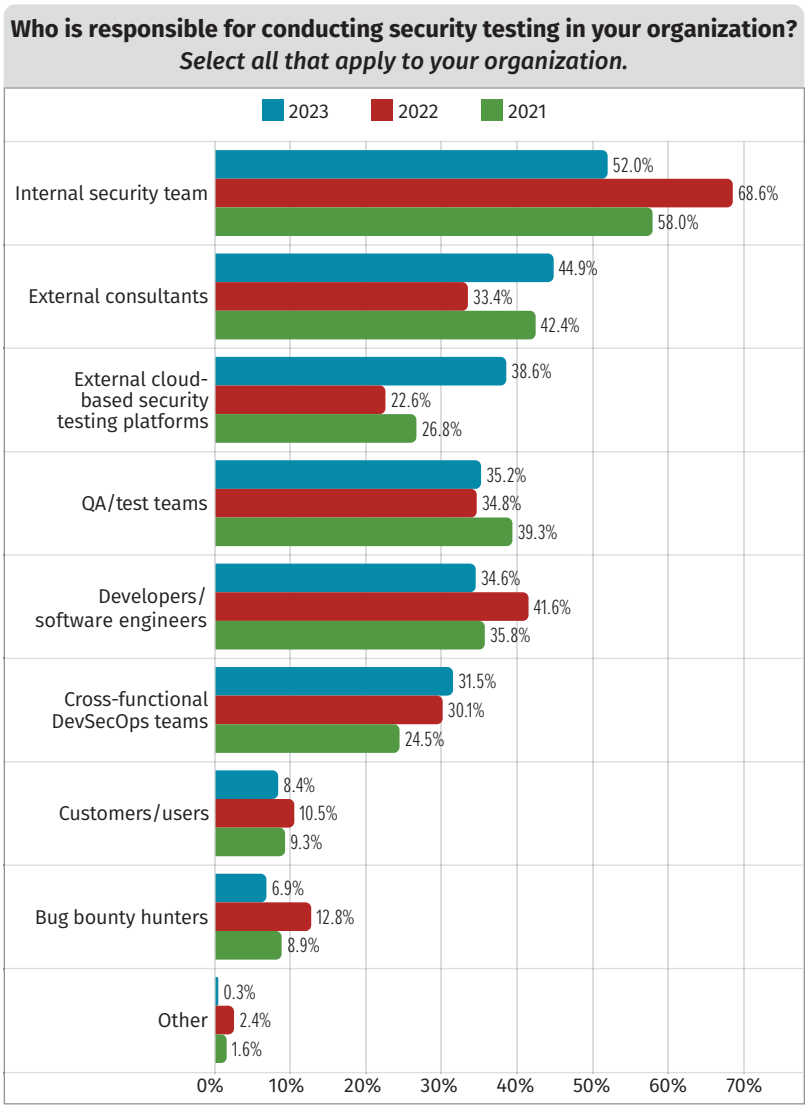

Figure 17. Stakeholders Conducting Security Testing, 2021–2023

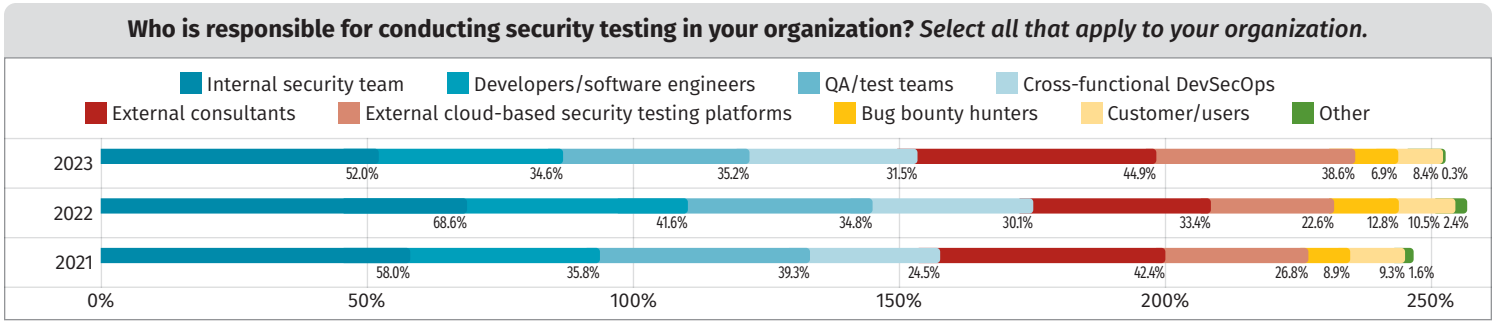

Figure 18. Security Testing by Role, 2021–2023

Where internal testing is concerned, there is still a heavy reliance on internal security teams. The survey responses indicate that organizations typically have two or more internal teams doing some manner of security testing and that nearly all have some external source for security findings (see Figure 18).

In the automated testing coverage chart (Figure 19), the responses from 2022 and 2023 are shown interleaved by year, with unknown and 0% coverage response percentages presented as negatives. This allows us to pick out three important changes reflected in the data:

- In every method represented, the total percentage of respondents covering 1% or greater of their codebase increased.
- In every method represented, the percentage of respondents in the 1–24% coverage category decreased.
- In every method represented, the total percentage of unknown or 0% responses decreased.

In other words, not only is the practice of testing expanding but the coverage is improving as well.

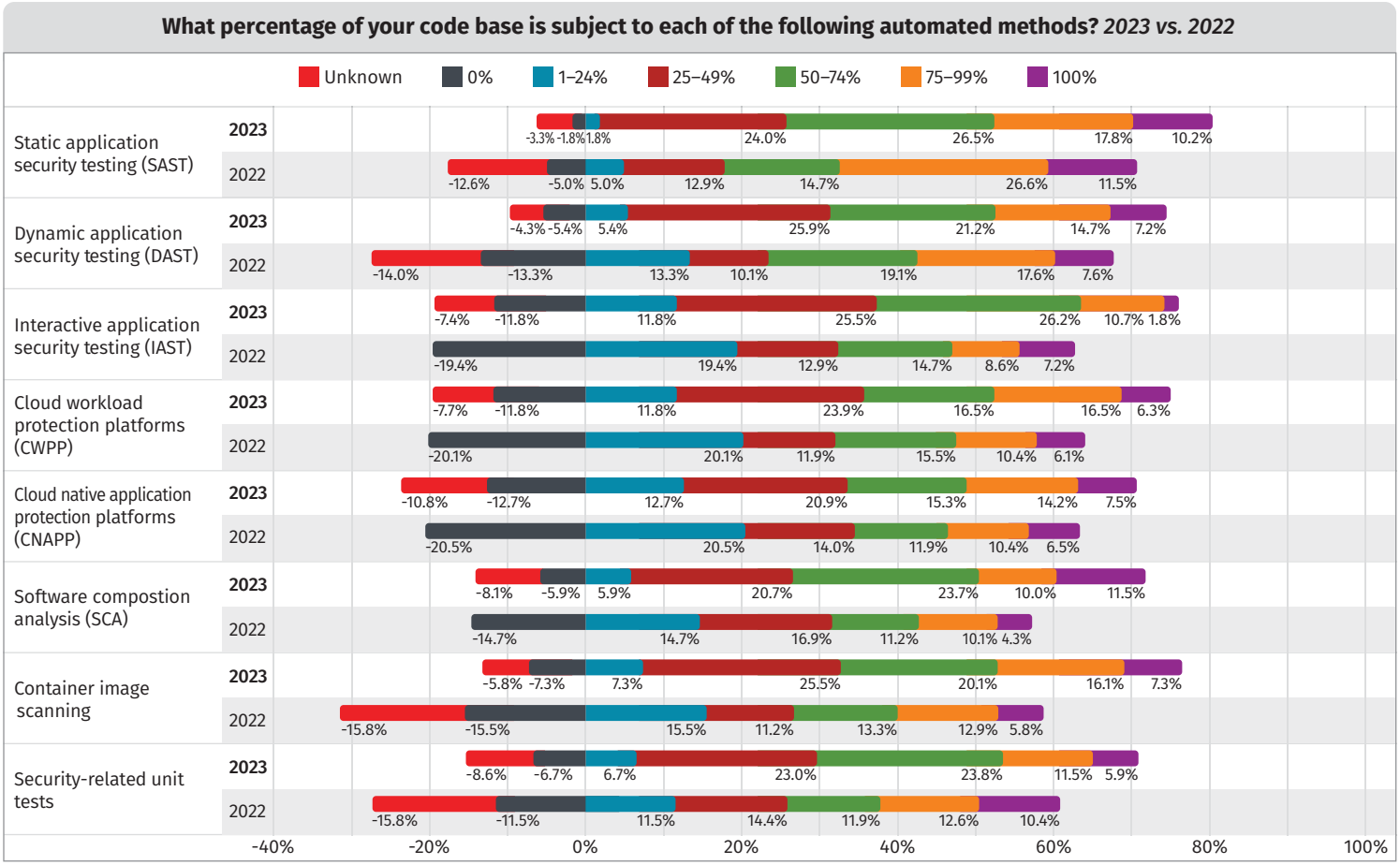

Figure 19. Code Base Subject to Automated Methods

This year we also see static application security testing (SAST) usage reported by over 85% of respondents, with 79% of respondents using SAST to cover at least 25% of their code. There was also a notable increase in software composition analysis (SCA) testing this year. Regardless of whether the SCA increase is related to the increase in SAST coverage, the increased adoption of cloud-hosted CI/CD platforms (which often include some form of SCA capability), or a blend of both, organizations are clearly taking supply chain security seriously. In light of the increased adoption of container-based cloud environments, the increased usage and coverage of container image scanning is also a positive sign.

TAKEAWAY

Before testing how an application behaves when running, it's crucial that organizations assess their supply chains, especially for containerized workloads. Scanning container images, analyzing the collection of third-party components used to build an application (SCA), and analyzing custom code with SAST tools all contribute to clearly understanding which risks are present in an application before it ever executes.

DevSecOps Tools and Practices: What Works?

Build automation, continuous integration, and automated testing remain the leading organizational practices, as they have been for the past two years. These are core practices for both DevOps and DevSecOps, so they will continue to be important areas for organizational investment.

Another continuing trend from prior years is that immutable infrastructure, blameless retrospectives, and chaos engineering remain underutilized practices (see Table 3).

Table 3. Respondents' Adoption Rates of Various DevSecOps Practices, 2021–2023

	Percentage			Ranking		
	2023	2022	2021	2023	2022	2021
Build automation	60.9%	83.3%	66.2%	1	1	1
Continuous integration (CI)	49.6%	57.2%	51.6%	2	3	2
Automated testing	44.9%	57.5%	51.6%	3	2	3
Microservices-based architecture	43.0%	51.9%	40.9%	4	5	5
Continuous deployment (CD) to production	42.7%	49.9%	35.6%	5	6	7
Automated deployment	41.0%	44.6%	43.1%	6	7	4
Programmable configuration management/infrastructure as code	39.9%	41.1%	33.8%	7	8	8
Continuous monitoring and measurement	32.0%	56.0%	35.6%	8	4	6
Immutable infrastructure provisioning	25.1%	24.3%	21.7%	9	9	9
Blameless retrospectives	11.8%	17.3%	11.7%	10	10	10
Chaos engineering	4.1%	9.1%	5.0%	11	11	11
Other	0.8%	2.9%	2.1%	12	12	12

Key Performance Indicators and Metrics

Many organizations collect a limited number of key performance indicators (KPIs) to establish long-term trends that can be used to identify aberrations and to forecast expectations in core business processes. KPIs can provide organizations the information needed to ensure stable DevSecOps processes, and they can provide insights into the impact of process and tool changes made within those processes.

The number of open security vulnerabilities remains the KPI that is most widely used to measure the success of DevSecOps programs, just as it was for the preceding two years. Time to fix security vulnerabilities was the second most important KPI in 2021 and 2022, but that KPI was displaced this year by false positive rates. This suggests that as programs mature and the volume of security test results increases, survey participants find that ensuring a finding is a true positive is more important than quickly addressing findings that may have little to no impact in their environments. The increase in the importance of false positive rates makes intuitive sense, when considered in conjunction with the overall increase in testing activity and coverage we previously mentioned (see Figure 19). Tracking coverage of automated tests rounds out the top 3 this year, just as it did last year (see Figure 20).

That these are the top 3 KPIs suggests that successful DevSecOps programs are using automated testing with wide coverage not only to verify changes to code but also to confirm that any reported vulnerabilities are true positives and to leverage the synergy between these capabilities to drive down the number of open security vulnerabilities.

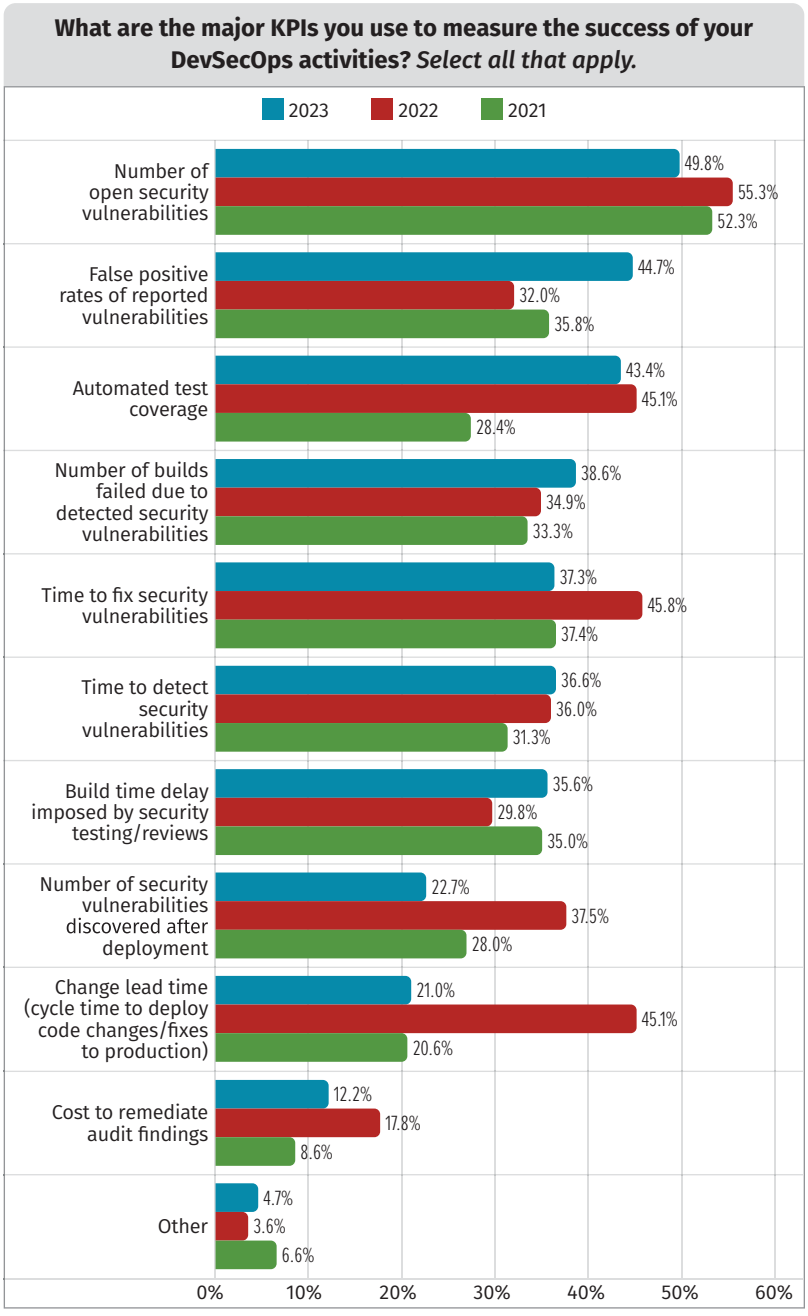

Figure 20. Top KPIs Used in Respondents' Organizations, 2021–2023

Top DevSecOps Challenges and Success Factors

The survey respondents' view of the No. 1 success factor in building a DevSecOps program has changed over the years, from training to buy-in to integrated security testing. This shift in the No. 1 success factor indicates that in just three short years organizations have made significant advances in learning to develop and implement DevSecOps programs. It also tells us that tooling is available to be integrated into DevSecOps practices. Communication, by contrast, has held second place continuously for the last three years SANS has conducted this survey.

Reinforcing the availability of tooling for DevSecOps teams and practitioners, this year's No. 1 challenge to implementing a solid program is the lack of sufficient budget for those tools, and for security programs overall. This has consistently placed as a top 5 challenge during the past three years.

The efforts to build successful DevSecOps programs—like any shift in organizational culture and practices implemented over multiple years—have been hampered by changing requirements and organizational silos. The mirrored stability of communication as a success factor (No. 2 for the past three years) and organizational silos as a challenge (No. 1 in 2021, No. 3 in 2022 and 2023) makes it clear that breaking down internal organizational barriers to enable communication remains fundamental to building a successful DevSecOps practice.

During the past three years, integrating automated security testing into workflows has risen from fifth to first as a success factor for organizations. Nobody who is familiar with the CALMS[6] model for DevOps—the C in CALMS stands for culture, the A for automation—will be surprised to see communication leading the way as an indicator of success in DevSecOps practices (see Table 4, on the next page).

TAKEAWAY

A successful DevSecOps program requires a strong focus on communication and culture to break down organizational silos. As the DevSecOps journey progresses, agreeing on the *why* (getting buy-in), understanding the *what* (training), and implementing the *how* (integrating tools into processes) may be temporary top priorities, but communication must not be neglected.

[6] CALMS Framework, www.atlassian.com/devops/frameworks/calms-framework

Table 4. Respondents' Ranking of DevSecOps Success Factors and Challenges, 2021–2023

Success Factor	Ranking			Maximum Rank	Percentage			Maximum Percentage
	2023	2022	2021		2023	2022	2021	
Integrating automated security testing into developer/engineering tool chains and build/deploy workflows	1	4	5	1	54.1	52.7	45.0	54.1
Improving communications across Dev, Ops, and security	2	2	2	2	52.0	55.6	51.4	55.6
Automating build/test/deploy/provisioning workflow, and thereby minimizing time/cost to fix vulnerabilities	3	3	7	3	51.4	55.3	43.4	55.3
Gaining developer/engineering buy-in	4	5	4	4	49.7	52.4	46.2	52.4
Gaining management buy-in	5	1	3	1	47.3	57.5	47.8	57.5
Sharing goals and measurable success factors across Dev, Ops, and security	6	8	5	5	35.8	29.8	43.4	43.4
Gaining security team buy-in	7	—	—	7	35.8	0.0	0.0	35.8
Training developers/engineers in secure coding	8	6	1	1	31.1	48.0	51.8	51.8
Creating cross-functional DevSecOps teams	9	10	9	9	30.1	27.3	35.1	35.1
Developing "security champions" in Dev and Ops teams	10	9	8	8	29.1	28.7	42.2	42.2
Enforcing security/compliance policies in code using programmable/immutable infrastructure	11	12	12	11	28.0	21.8	20.7	28.0
Defining success clearly and measurably (e.g., metrics)	12	7	11	7	26.0	36.0	26.7	36.0
Reusing "secure by default" frameworks, libraries, templates, and services	13	11	10	10	25.0	25.8	31.5	31.5
Following a common compliance framework	14	13	13	13	4.1	6.2	10.0	10.0
Challenges	**Ranking**			**Maximum Rank**	**Percentage**			**Maximum Percentage**
	2023	2022	2021		2023	2022	2021	
Insufficient budget/funding for security programs and tools	1	4	2	1	46.8	38.4	48.8	48.8
Continuously changing requirements and priorities	2	6	4	2	41.9	32.3	36.6	41.9
Organizational silos between Dev, Ops, and security	3	3	1	1	41.5	43.4	50.0	50.0
Lack of developer/engineer buy-in	4	2	5	2	38.5	44.1	34.3	44.1
Shortage of application security personnel/skills	5	1	3	1	37.2	44.1	37.8	44.1
Lack of transparency into development/operations work	6	7	6	6	36.2	31.2	28.7	36.2
Lack of management buy-in	7	5	10	5	29.6	35.1	25.2	35.1
Lack of security team buy-in	8	9	12	8	27.2	24.4	20.5	27.2
Shortage of cloud engineering personnel/skills	9	8	7	7	26.9	18.6	28.3	28.3
Shortage of cloud security personnel/skills	10	13	8	8	24.9	25.1	26.4	26.4
Lack of coding skills in security teams	11	11	13	11	23.9	22.9	19.7	23.9
Inadequate/ineffective security training for developers/engineers	12	10	11	10	20.9	23.3	24.4	24.4
Inadequate test automation/over-reliance on manual testing	13	14	19	13	16.3	16.8	13.0	16.8
Compliance risks or lack of compliance buy-in	14	16	15	14	15.6	15.4	16.5	16.5
Security testing/scanning tools are inaccurate/unreliable	15	18	17	15	15.3	11.0	14.2	15.3
Lack of security tool support for languages, frameworks, and platforms	16	17	16	16	13.6	12.2	15.0	15.0
Security testing/scanning tools are too noisy and do not help prioritize resolution (e.g., exposure, exploitability, criticality)	17	19	18	17	12.0	9.7	13.4	13.4
Supply chain risks in third-party/open source components, APIs, and containers	18	15	20	15	9.3	15.4	12.6	15.4
Technical debt and security debt in legacy system environments	19	12	14	12	8.0	22.6	18.1	22.6
Security testing/scanning tools are too slow to fit into rapid release cycles/continuous deployment	20	20	21	20	7.6	7.9	7.1	7.9
Security capabilities of cloud platforms are inadequate	21	21	22	21	6.6	3.9	6.3	6.6

Future Trends

One of the notable forward-looking trends the 2023 survey shows is a significant increase (16%) in the use of AI or data science to improve DevSecOps through investigation and experimentation—from 33% in 2022 to 49% in 2023. The intense recent publicity about AI and the increasing availability of AI models, training data, and tools make this an area where ongoing adoption seems highly likely. That said, a strong contingent of the respondents (approximately 30%) reported not using AI or data science capabilities at all. This may reflect issues such as the rising level of concern surrounding data privacy and ownership of intellectual property. Responses also captured an increase in pilot projects integrating security operations into both the "AI and machine learning ops" and "data science operations" categories—a possible indication that organizations are performing threat modeling and risk assessments prior to incorporating AI capabilities into products (see Table 5).

Table 5. Emerging Technology for DevSecOps

Success Factor	Unknown	Not at All	Conducting Preliminary Investigation	Experimenting or Conducting Pilot Projects	Partially Integrated	Fully Integrated
Applying artificial intelligence or data science to improve DevSecOps	10.8%	28.6%	33.8%	15.3%	5.9%	4.5%
Integrating security operations into artificial intelligence/machine learning ops	9.4%	20.9%	26.5%	22.0%	12.5%	5.9%
Integrating security operations into data science ops	8.4%	19.9%	22.0%	24.4%	17.8%	6.3%
Utilizing serverless computing to build, manage, and scale applications	5.9%	11.1%	25.4%	24.7%	19.2%	11.1%
Leveraging GitOps to test, verify, automate, deploy, and advance/mature the principles of infrastructure as code	11.5%	15.0%	19.2%	18.1%	22.3%	12.5%
Developing with microservices rather than monolithic applications to improve the overall agility and flexibility for DevSecOps	9.1%	17.1%	22.0%	20.6%	20.6%	10.5%
Leveraging application security orchestration and correlation (ASOC) tools for DevSecOps	12.5%	24.0%	21.3%	15.0%	16.7%	8.7%
Pursuing a platform engineering approach to streamline application development, analysis, deployment, and operations	10.5%	19.5%	21.6%	19.5%	17.4%	9.4%

FaaS, GitOps, and microservices share both the most overall attention from organizations and the largest percentages of full integration. These three practices are often interrelated and interwoven, so it makes sense that they move as a group, reinforcing one another.

The subject of platform engineering to streamline the flow from idea to implementation was added to the survey this year. The responses show greater awareness and adoption of platform engineering practices than of application security orchestration and correlation (ASOC) tools. As the developer self-service features inherent in a platform engineering practice mature, it will be essential to leverage the orchestration used to build, package, test, and deploy an application to incorporate security testing and tooling at key points along the path that has been laid out. A well-implemented software engineering platform, designed in close collaboration with security stakeholders, could likely meet an organization's ASOC objectives.

This year's respondents replied to the overall set of Future Trends questions with an unknown response roughly half as often as last year. It seems possible that this indicates that DevSecOps efforts are being communicated better within organizations, indicating that the sharing principle from the CALMS framework (the S in CALMS) is taking hold.

From 2022 to 2023, all positive responses increased, with the exception of "developing with microservices," which saw a small decrease (see Figure 21).

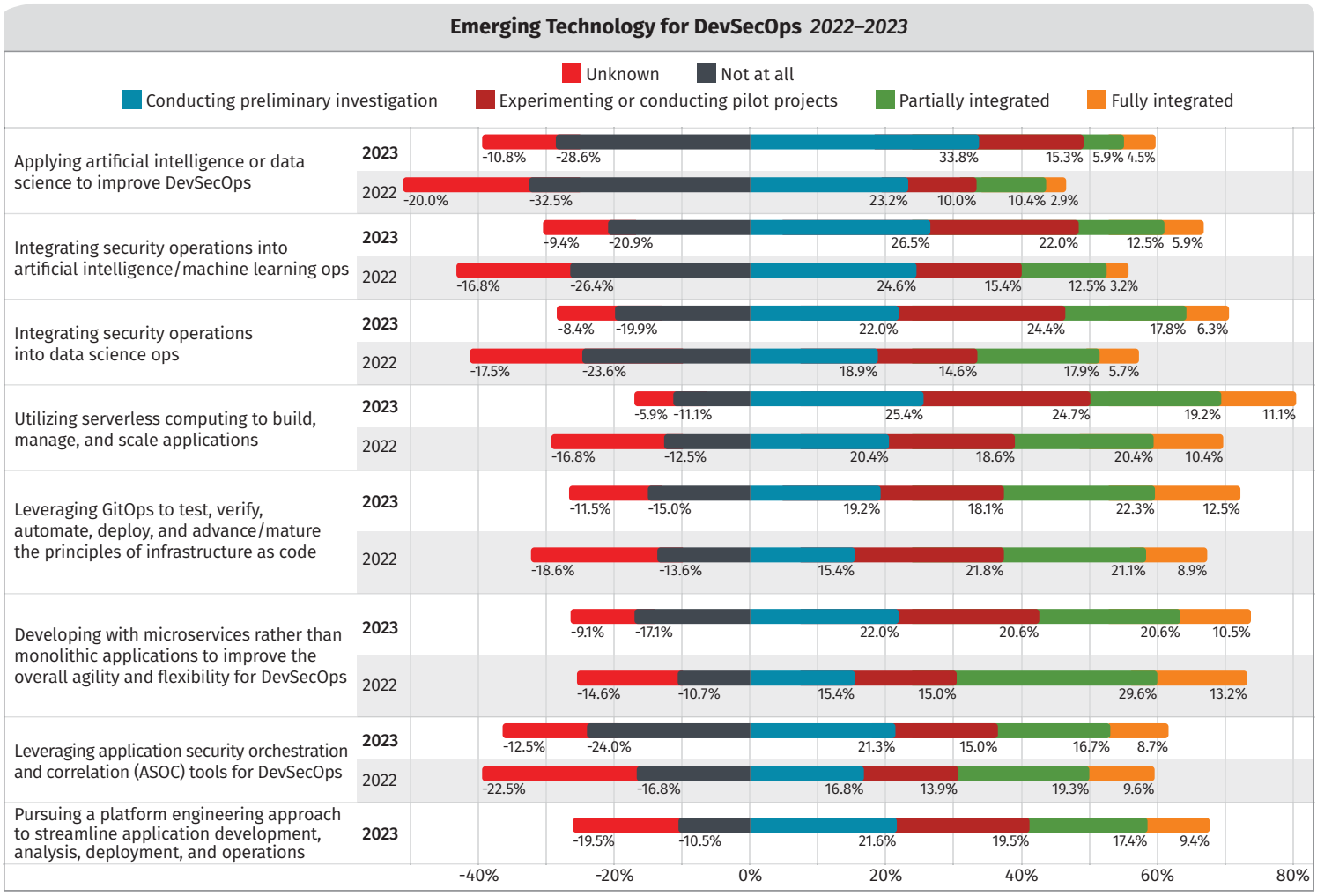

Figure 21. Emerging Technology for DevSecOps, 2022–2023

Going Forward

A DevSecOps program needs to integrate the practices of security, development, and operations teams, creating a cohesive, collaborative system development life cycle. This requires substantial initial and ongoing investment by the organization, but its benefits—which include reduced time to fix security issues, less burdensome security processes, and increased ownership of application security[7]—are well understood.

This year's survey continues to point toward increasing maturity and adoption of DevSecOps practices, but the survey data also reveal areas for improvement. Key takeaways from the 2023 survey include:

- Although workloads are migrating to the cloud, DevSecOps teams may be missing out on some of the advantages of immutable containers and ephemeral serverless functions. Both approaches fit well with CI/CD deployments and can be utilized to create applications that are secure, performant, and potentially more cost-effective than VMs.
- Multicloud has become the norm. When organizations use multiple cloud providers, the work to secure those clouds grows exponentially. DevSecOps practitioners should consider implementing and expanding the use of open source or commercial CSPM tools to assess and manage infrastructure security at scale. Additionally, using CWPPs can enable organizations to protect resources across cloud providers.
- DevSecOps teams should continue to invest in tools that help to ensure the security and integrity of their applications and all the dependent components in their software supply chains.
- Organizations should leverage KPIs to identify the most important area for the organization to improve next. Benchmarking against peer organizations' metrics can be used to expand management support, and they also help to demonstrate due care.
- DevSecOps teams should limit the programming languages approved for new development projects based on security risks and availability of security testing tools (among other factors), and they should refactor older code written with memory-unsafe languages as opportunities arise.
- When moving workloads to the cloud, organizations must choose between a lift-and-shift approach that minimizes the use of cloud-provider-specific capabilities and a rebuild-and-integrate approach that makes intentional use of each cloud provider's unique capabilities. There is no one-size-fits-all approach, so organizations should develop guidance to apply consistently to their decision-making process.
- Organizations should continue to champion a culture of communication and shared responsibility for security across teams, processes, and projects.
- As machine learning and AI efforts erupt across organizations, they should continue to apply the "shift security left" mentality by performing risk assessments and creating threat models for AI experiments and projects before starting work.

[7] DevOps Digest, "A Primer on Secure DevOps: Learn the Benefits of These 3 DevSecOps Use Cases," July 18, 2022, www.devopsdigest.com/secure-devops-use-cases

Organizations continue to be pressured to do more work with fewer resources—especially personnel. DevSecOps is an approach that can help relieve some of that pressure. The right KPIs will keep teams focused on the proper priorities, and investments in automation for build, test, and deployment work will increase agility, including agility in responding to security incidents.

Critical focus areas for a successful DevSecOps program are:

- Early consideration for the security facets of any project, through risk assessment and threat modeling prior to writing any code
- Automation of security tests aligned with well-defined standards and practices
- Comprehensive understanding of the security status of resources required to run your applications, including infrastructure, third-party software and software developed in-house
- Automation of the entire build, test, and deploy process, to accelerate responding to attacks and vulnerabilities and to enable automatic remediation

Many organizations feel an urgent need for more qualified DevSecOps personnel. Because demand continues to outweigh supply in this area, there is a real need to spark more interest in this ever-changing field. To cope with the scarcity of talent amid competitive pressures, organizations should further leverage proven DevSecOps practices and explore emerging technological capabilities. This may mean harnessing some of the underutilized technology (for example, CSPM, CWPP, AI, machine learning ASOC) or applying new tools, technologies, and practices (for example, immutable infrastructure, zero trust, benchmarking) in pursuit of optimizing and streamlining DevSecOps.

This survey showcases the progress made by the DevSecOps community in improving organizations' security postures and organizational effectiveness, recognizes the challenges it still faces, and highlights areas for additional focus on the path to DevSecOps excellence.

Sponsor

SANS would like to thank this survey's sponsor:

sonatype