ESG | Enterprise Strategy Group | Getting to the bigger truth.™

RESEARCH HIGHLIGHTS

Modern Application Development Security

Dave Gruber, Senior ESG Analyst

AUGUST 2020

ESG

CONTENTS

Research Objectives

DevSecOps has moved security front and center in the world of modern development; however, security and development teams are driven by different metrics, making objective alignment challenging. This is further exacerbated by the fact that most security teams lack an understanding of modern application development practices. The move to microservices-driven architectures and the use of containers and serverless has shifted the dynamics of how developers build, test, and deploy code.
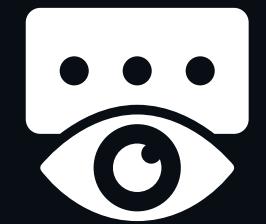
As a result, a convergence of application security tools is underway. Organizations are overwhelmed with the amount of and overlap in issues raised from multiple testing tools, complicating prioritization and mitigation, so integrated application security platforms are desired.

In order to gain insight into these trends, ESG surveyed 378 IT, cybersecurity, and application development professionals at organizations in North America (US and Canada) involved with securing application development tools and processes.
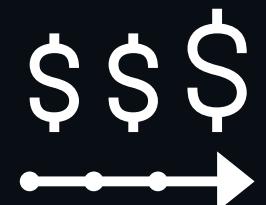
THIS STUDY SOUGHT TO:
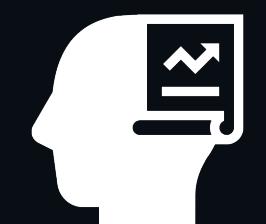
Examine the buying intentions of application security teams regarding dev-time application security controls and gauge buyer preferences for different types of vendors' application security solutions.

Determine the extent to which security teams understand modern development and deployment practices, and where security controls are required to mitigate risk.

Understand the trigger points influencing application security investments and how decision makers are prioritizing and timing purchasing decisions.

Gain insights into the dynamics between development teams and cybersecurity teams with respect to the deployment and management of application security solutions.

Research Highlights

Most think their application security programs are solid, though many still push vulnerable code.

Having a good application security program doesn't mean that organizations don't still push vulnerable code. The difference is that those that push such code do so knowingly and with a thorough understanding of the risks that they are taking. Application security requires a constant triage of potential risks, involving prioritization decisions that allow development teams to mitigate risk while still meeting key deadlines for delivery.

Multiple security testing tools are needed to secure the potpourri of application development and deployment models in use today.

As application security has matured, no single testing technique has helped development teams mitigate all security risk. This requires teams to employ multiple tools, often from multiple vendors, to secure the SDLC. Usage varies, as do the tools that organizations deem most important, but most organizations end up utilizing a "cocktail" of tools to satisfy their security needs.

Developer security training is spotty, and programs to improve developer security skills are lacking.

While most provide developers with some level of security training, more than 50% only do so annually or less often. While development managers are often responsible for this training, in many organizations, application security analysts carry the burden of performing remedial training for development teams or individual developers who have a track record of introducing too many security issues.

The proliferation of AppSec testing tools is an issue for many, with more than a third focusing investments on consolidation.

Like other categories of security controls, many organizations are employing so many tools that they are struggling to integrate and manage them. This reduces the effectiveness of the program and directs an inordinate amount of resources to managing tools. Nearly one-third of organizations are experiencing this issue and are therefore planning future investment to consolidate and simplify their tools proliferation.

Organizations are investing, with more than half planning to significantly increase spending on application security over the prior year.

While 44% plan on targeting application security investments toward cloud, one-third are focused on consolidation of tools to simplify the process. Others plan to invest in expanding the use of testing tools to a higher percentage of their development teams and applications.

Most think their application security program is solid, though many still push vulnerable code.
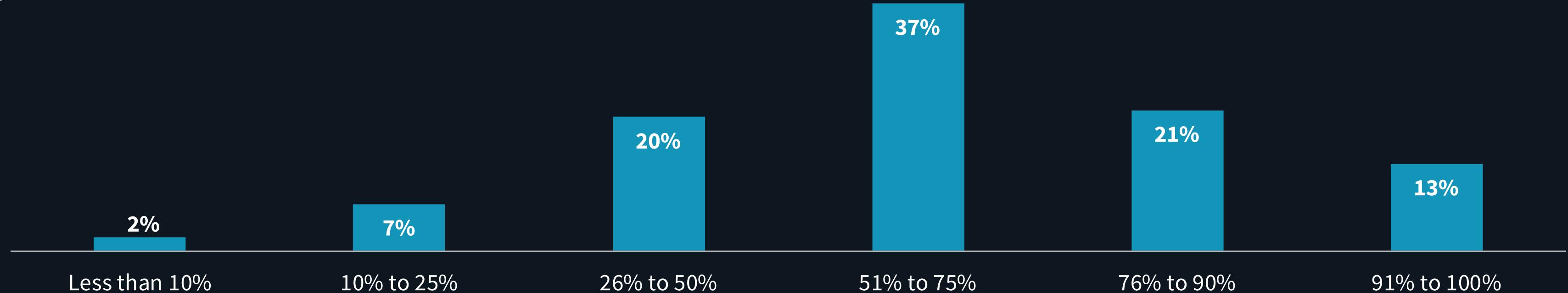
Most Think Their Application Security Program Is Pretty Good

Most organizations think that their application security programs are pretty good, with more than a third providing a rating of 9 or 10, and an overall mean of 7.92. This favorable assessment reflects continued investment and coverage levels in application security programs over the past few years. Still, code coverage is far from complete, with only 34% using AppSec tools on more than three-quarters of their codebase.

36%
give their application security programs a rating of 9 or 10

Codebase protected by AppSec tools

Less than 10%	10% to 25%	26% to 50%	51% to 75%	76% to 90%	91% to 100%
2%	7%	20%	37%	21%	13%
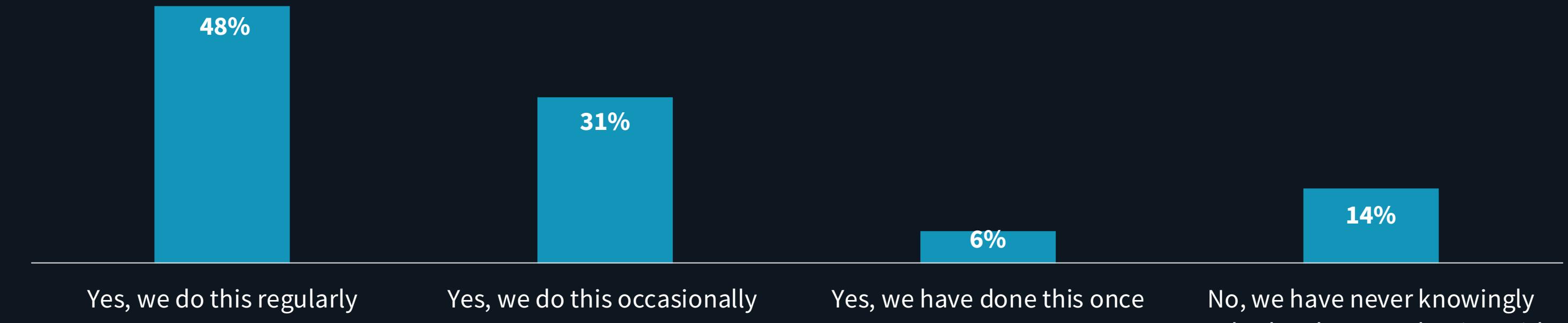
Despite Good Programs, Most Still Regularly Push Vulnerable Code

Having a good application security program doesn't mean that organizations don't still push vulnerable code. The difference is that those that push such code do so knowingly and with a thorough understanding of the risks that they are taking. Application security requires a constant triage of potential risks, involving prioritization decisions that allow development teams to mitigate risk while still meeting key deadlines for delivery. Note that vulnerabilities discovered too late in the cycle often don't get mitigated, reinforcing the importance of shifting application security as far left as possible to leave enough runway to resolve critical issues in time for delivery.

Do organizations push vulnerable code?

48%	Yes, we do this regularly
31%	Yes, we do this occasionally
6%	Yes, we have done this once
14%	No, we have never knowingly pushed code to production with vulnerabilities

Why organizations push vulnerable code

To meet a critical deadline, with a plan to remediate in a later release	54%
We felt the vulnerabilities had very low risk	49%
The vulnerabilities were discovered too late in our cycle to resolve them in time	45%

Most Still Experience Exploits

While increased investment in AppSec programs is reducing risk, 60% still report exploits from OWASP top-10 vulnerabilities. These exploits aren't necessarily tied to code where known vulnerabilities were identified, but this highlights the diligence that is required, including code coverage, testing frequency throughout the SDLC, and prioritization of identified vulnerabilities.

So who makes the decision to ship code with known vulnerabilities? While many share this decision between a development manager and the security analyst, many teams leave the final decision to a single, responsible person. This reflects a variation in how different development organizations manage the overall application security process, with some putting ownership on the security team while others hold the development manager accountable.

60%
have had production applications exploited by OWASP top-10 vulnerabilities in the past 12 months
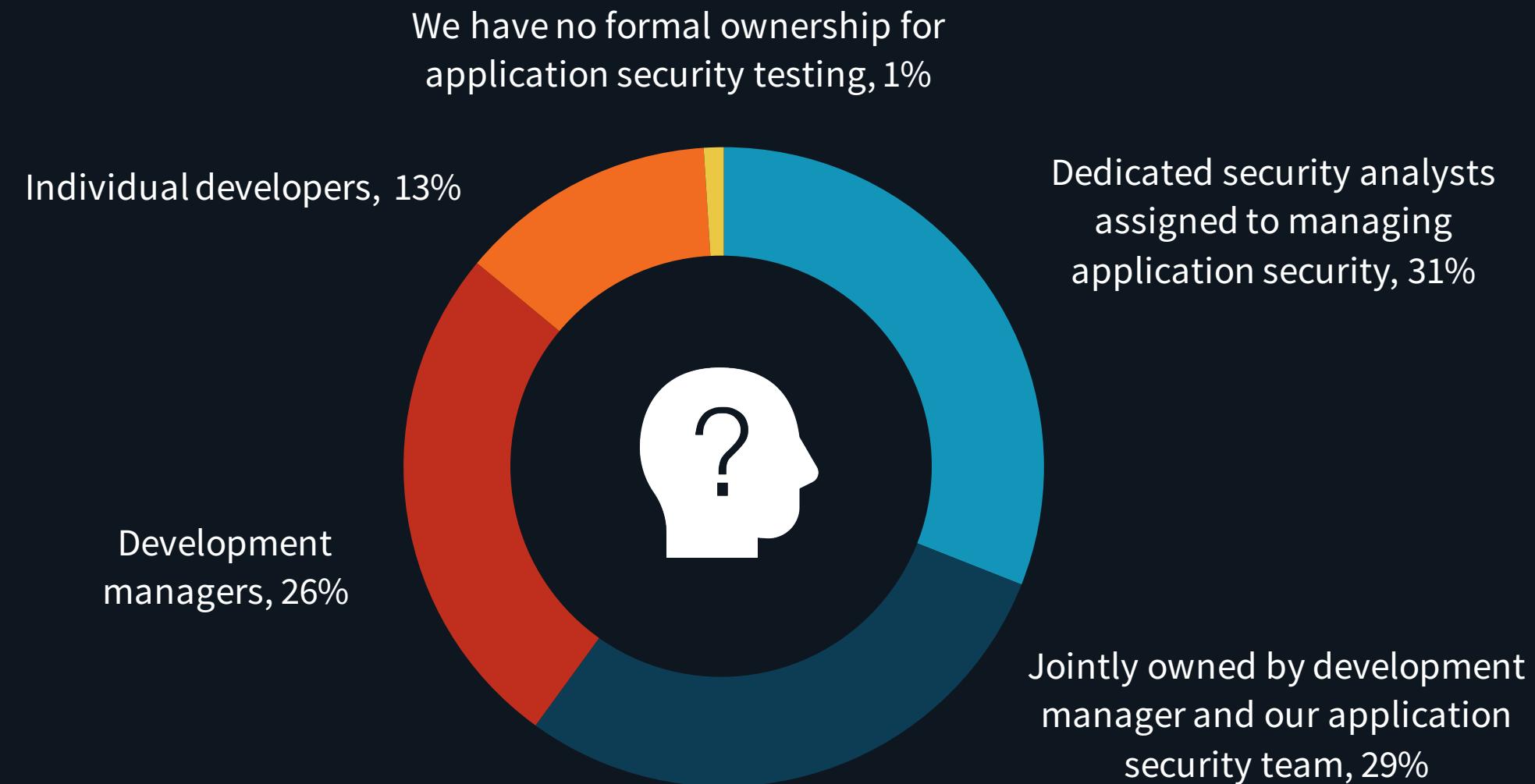
Who makes the decision to push code?

- Team decision that includes both development manager and security analyst
- Development manager
- Security analyst
- Individual developers assess the priority of each issue
- QA and/or security teams
- Don't know

28%
24%
21%
15%
11%
1%
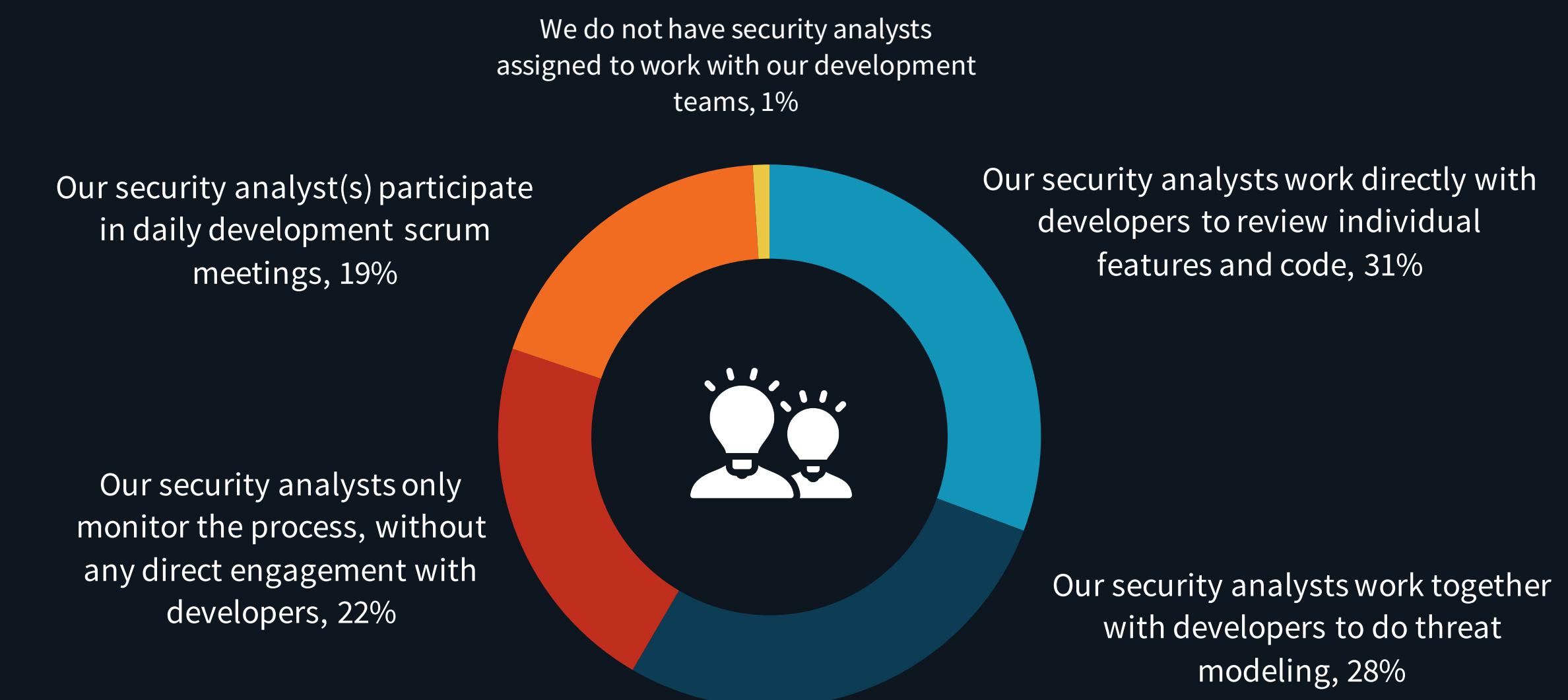
Security Analysts Are Increasingly Involved with AppSec Testing

In most organizations, the development manager or the application security analyst is responsible for the testing program, while 29% of organizations report that they jointly are responsible for the program. Security analysts play a major role in helping developers build applications securely. 78% of organizations report that their security analysts are directly engaged with their developers: 31% work directly with developers to review individual features and code, 28% work with developers to do threat modeling, and 19% participate in daily scrums. This high level of engagement drives both learning and oversight to ensure applications are built securely.

Application security testing ownership

- We have no formal ownership for application security testing, 1%
- Individual developers, 13%
- Dedicated security analysts assigned to managing application security, 31%
- Development managers, 26%
- Jointly owned by development manager and our application security team, 29%

How security analysts are helping to improve programs

- We do not have security analysts assigned to work with our development teams, 1%
- Our security analyst(s) participate in daily development scrum meetings, 19%
- Our security analysts work directly with developers to review individual features and code, 31%
- Our security analysts only monitor the process, without any direct engagement with developers, 22%
- Our security analysts work together with developers to do threat modeling, 28%

10 Elements of the
Most Effective
AppDev Security Programs

1. Application security controls are highly integrated into the CI/CD toolchain.

2. Application security best practices are formally documented.

3. Application security training is included as part of the ongoing development security training program.

4. Development managers are responsible for communicating best practices to developers.

5. A high percentage of developers participate in formal application security training programs.

6. Security issue introduction is tracked for individual development teams.

7. Formal processes and metrics track continuous improvement of application security.

8. Continuous improvement metrics are tracked for individual development teams.

9. Security issues are tracked during the code development process.

10. Automated risk aggregation tools roll up risk to keep senior development leaders informed.
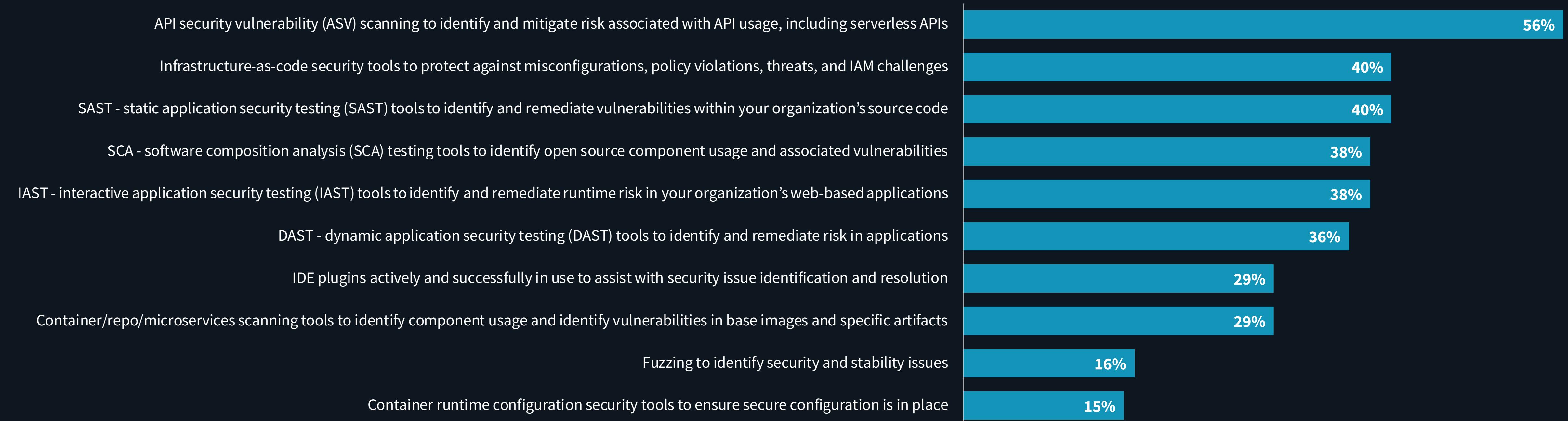
Multiple security testing tools are needed to secure the potpourri of application development and deployment models in use today.

A Wide Variety of AppSec Testing Tools Are in Use

As application security has matured, no single testing technique has helped development teams mitigate all security risk. This requires teams to employ multiple tools, often from multiple vendors, to secure the SDLC. Usage varies, as do the tools that organizations deem most important, but most organizations end up utilizing a "cocktail" of tools to satisfy their security needs.
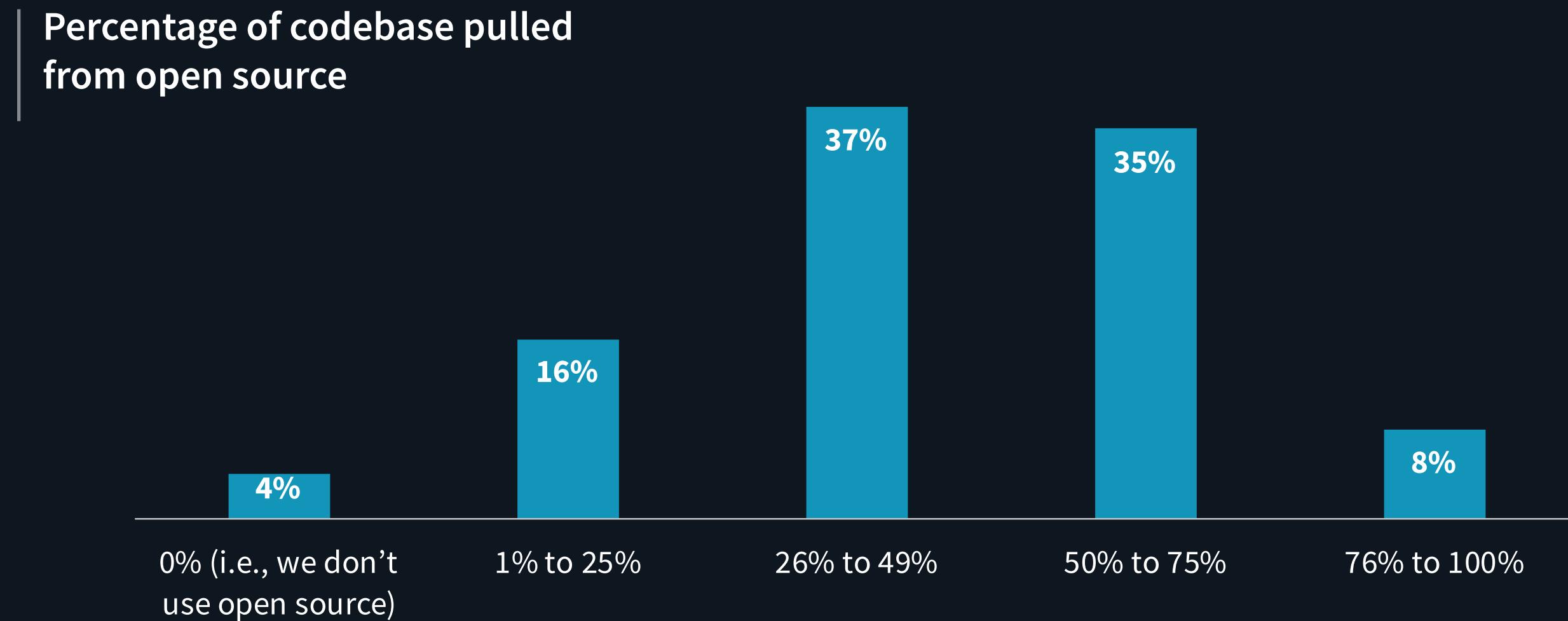
As new development and deployment models emerge, new testing tools emerge to secure them. Some matriculate back into broader testing platforms while others stand alone for the long haul.

Application security testing ownership

API security vulnerability (ASV) scanning to identify and mitigate risk associated with API usage, including serverless APIs	56%
Infrastructure-as-code security tools to protect against misconfigurations, policy violations, threats, and IAM challenges	40%
SAST - static application security testing (SAST) tools to identify and remediate vulnerabilities within your organization's source code	40%
SCA - software composition analysis (SCA) testing tools to identify open source component usage and associated vulnerabilities	38%
IAST - interactive application security testing (IAST) tools to identify and remediate runtime risk in your organization's web-based applications	38%
DAST - dynamic application security testing (DAST) tools to identify and remediate risk in applications	36%
IDE plugins actively and successfully in use to assist with security issue identification and resolution	29%
Container/repo/microservices scanning tools to identify component usage and identify vulnerabilities in base images and specific artifacts	29%
Fuzzing to identify security and stability issues	16%
Container runtime configuration security tools to ensure secure configuration is in place	15%

Modern Codebases Are Heavily Dependent on Open Source, But Less Than Half Are Currently Using Open Source Security Controls

While different types of testing tools have been around for years, adoption is still not where it should be. For example, while the use of open source software in modern application development is significant, still less than half of development teams report currently utilizing open source security testing tools. While many have plans to do so, this unsettling trend demonstrates the current state of application security testing adoption in many companies today.

Percentage of codebase pulled from open source

| Category	Percentage
0% (i.e., we don't use open source)	4%
1% to 25%	16%
26% to 49%	37%
50% to 75%	35%
76% to 100%	8%
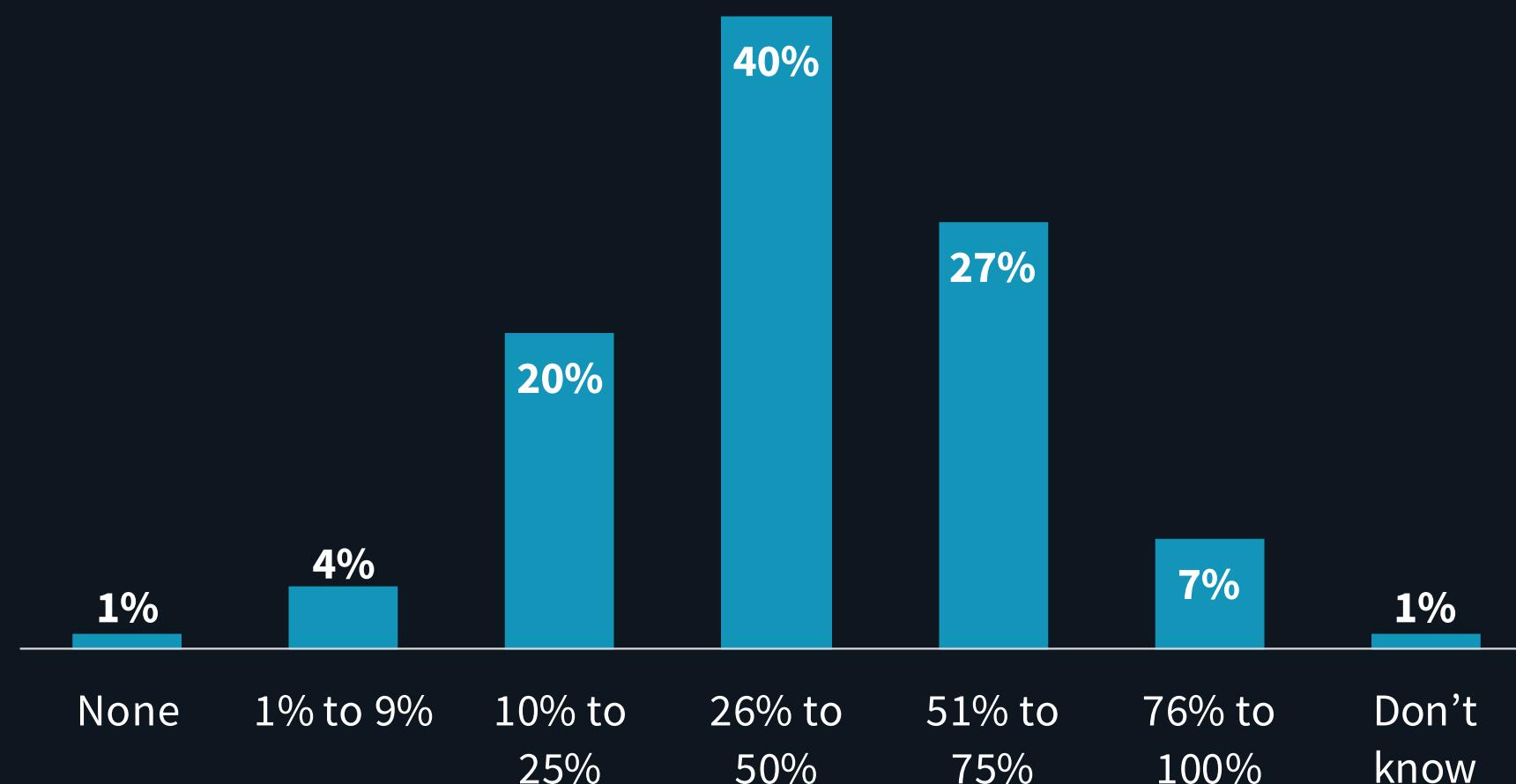
48%
have already invested in specific security controls to scan for open source vulnerabilities.
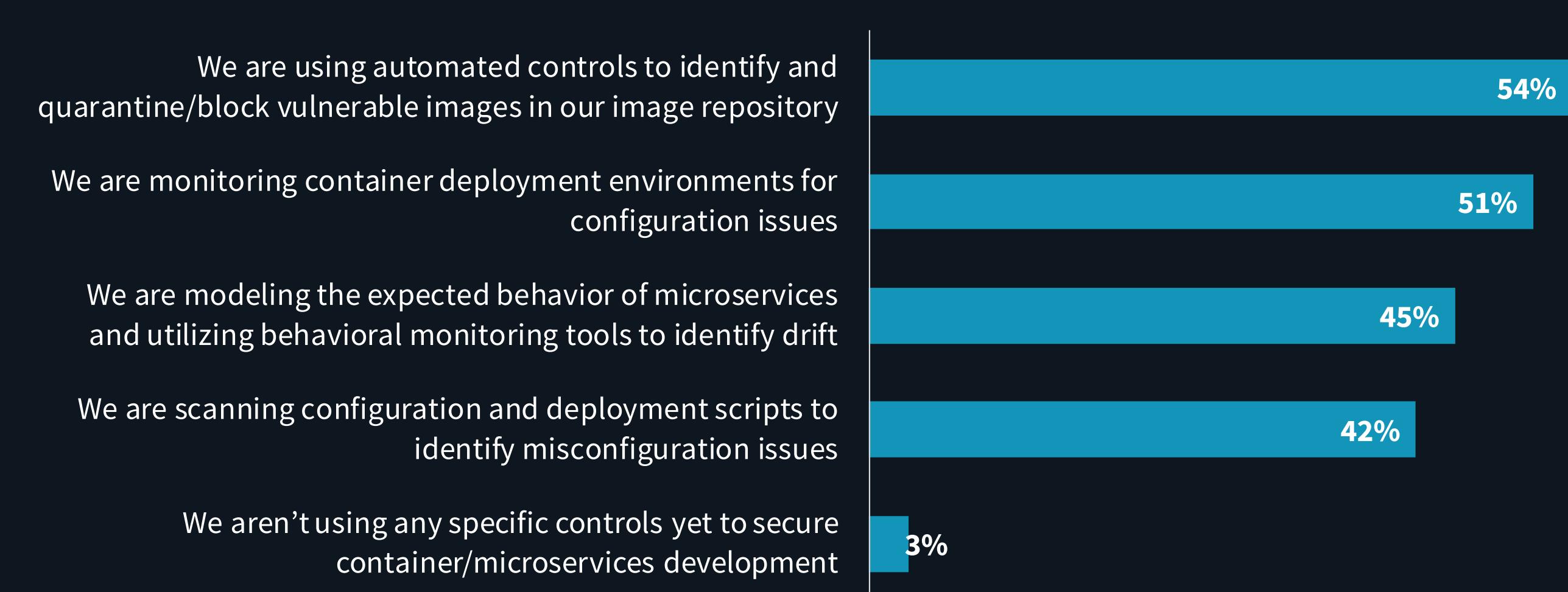
Microservices Container Development Is Gaining Steam

In some cases, more modern development and deployment models get security attention sooner. Here we see that microservices container development has become widely adopted in a relatively short period of time, and that the use of specific security controls has followed along with it. Other cloud development and deployment models are seeing similar patterns of adoption.

Percentage of development teams using containers

- None: 1%
- 1% to 9%: 4%
- 10% to 25%: 20%
- 26% to 50%: 40%
- 51% to 75%: 27%
- 76% to 100%: 7%
- Don't know: 1%

Controls in place to secure containers

- We are using automated controls to identify and quarantine/block vulnerable images in our image repository: 54%
- We are monitoring container deployment environments for configuration issues: 51%
- We are modeling the expected behavior of microservices and utilizing behavioral monitoring tools to identify drift: 45%
- We are scanning configuration and deployment scripts to identify misconfiguration issues: 42%
- We aren't using any specific controls yet to secure container/microservices development: 3%

Challenges with Current Testing Tools

Ultimately, it's up to developers to mitigate identified security issues; however, many report that the most common challenge with their current tools is that their developers lack the knowledge to do so. Security tools vendors provide guidance through just-in-time training or recommended fixes, but the developer ends up with the work. Issue mitigation is often tied to better understanding how and why certain code introduces issues, so developer security training should gradually address this issue.

Others struggle with integrating with other AppSec tools, while many still worry about the friction that AppSec adds, slowing the overall development process.

Top challenges with current testing tools

Challenge	Percentage
Developers lack the knowledge to mitigate issues identified	29%
Difficulty or lack of integration between different application security vendor tools	26%
Adds friction and slows down our development cycles	26%
Developers are not utilizing tools we've invested in effectively	24%
Lack of ability to aggregate and dedupe findings from the various security tools	24%
Poor integration with development/DevOps tools	23%
Lack of a centralized reporting and management dashboard/console for vulnerability management	20%
Scans too slow	18%
Too many false positives	15%
Poor automation support	15%
Too many false negatives	14%

DevOps Integration Is Important to Improving Application Security Programs

Most believe that automating application security testing throughout the SDLC can make the biggest impact on the success of their program. DevOps integration reduces friction and shifts security further left, helping organizations identify security issues sooner. While developer education and improved tools and processes will no doubt also improve programs, automation is central to modern application development practices.
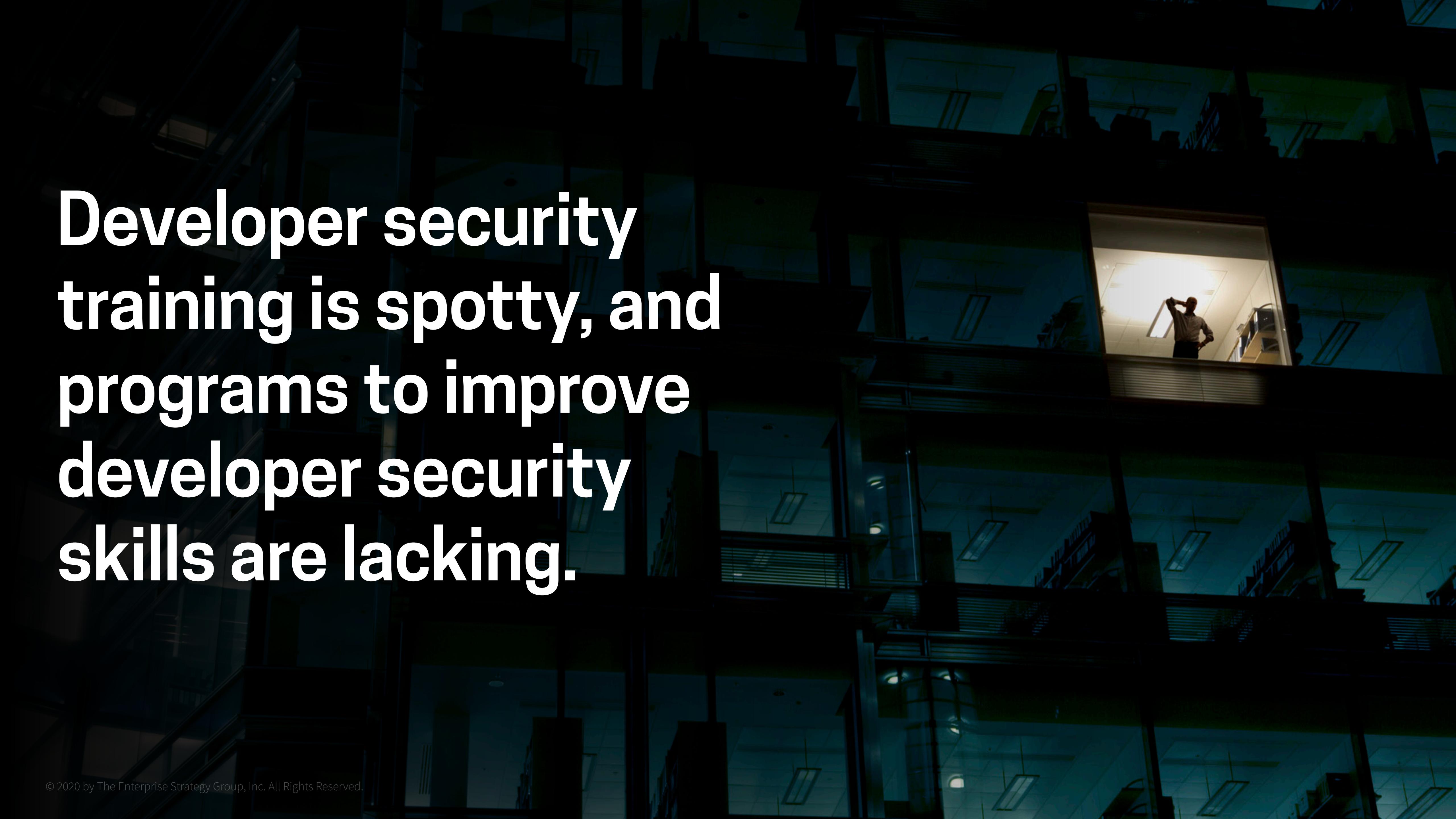
Level of DevOps and AppSec integration

- 5%
- 1%
- 38%
- 56%

- We utilize a highly integrated set of security controls throughout our DevOps process
- We use selective controls, but continue to invest in integrating additional controls
- Our application security tools are not well integrated into our processes
- We are pushing security as far left as possible in our processes

43%
believe DevOps integration is most important to improving AppSec programs.
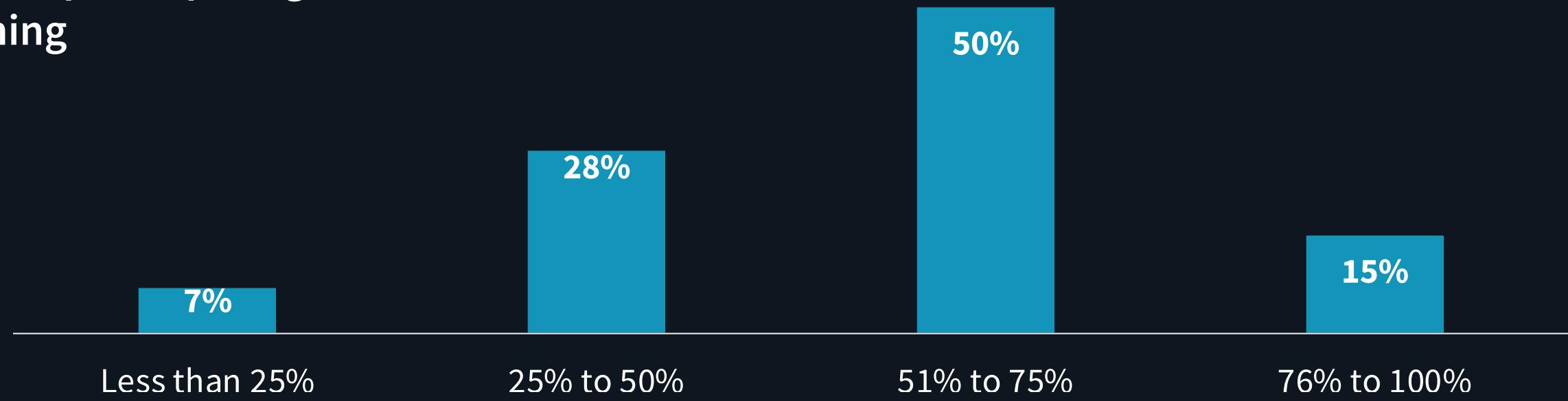
# Developer security training is spotty, and programs to improve developer security skills are lacking.

Most Require Developers to Consume AppSec Training

Most organizations require their developers to consume some amount of application security training, but 35% say that less than half of their development teams are participating in formal training. Only 15% report that all their developers are participating. As for frequency, less than half require their developers to engage in formal training more than once per year.

Percentage of developers participating in formal security training

7% — Less than 25%
28% — 25% to 50%
50% — 51% to 75%
15% — 76% to 100%

Security training requirements for application developers

We have no security training for developers, 1%

Developers are expected to self-educate on application security, 16%

Developers are only required to participate in training on an annual basis, 17%

Our developers utilize just-in-time training available from within our security tools, 17%

Developers are required to participate in training at least once per quarter, 29%

We only provide security training when developers join the team, 20%

Most Lack Programs to Measure the Effectiveness of Developer Security Training

Continuous improvement of security programs requires the measurement of issue introduction by development teams and individual developers. Slightly more than 40% report tracking both issue introduction and continuous improvement metrics, enabling targeted efforts to improve those teams and individuals who introduce the most issues.

How security training efficacy for application development teams is measured

Security issue introduction is tracked for each of our development teams	42%
Continuous improvement metrics are tracked for each development team	41%
Continuous improvement metrics are tracked for each developer	38%
Testing from within our training tools	37%
Issue introduction is tracked for each developer	32%
Issue introduction is tracked at the company level	28%
We don't measure training efficacy	2%
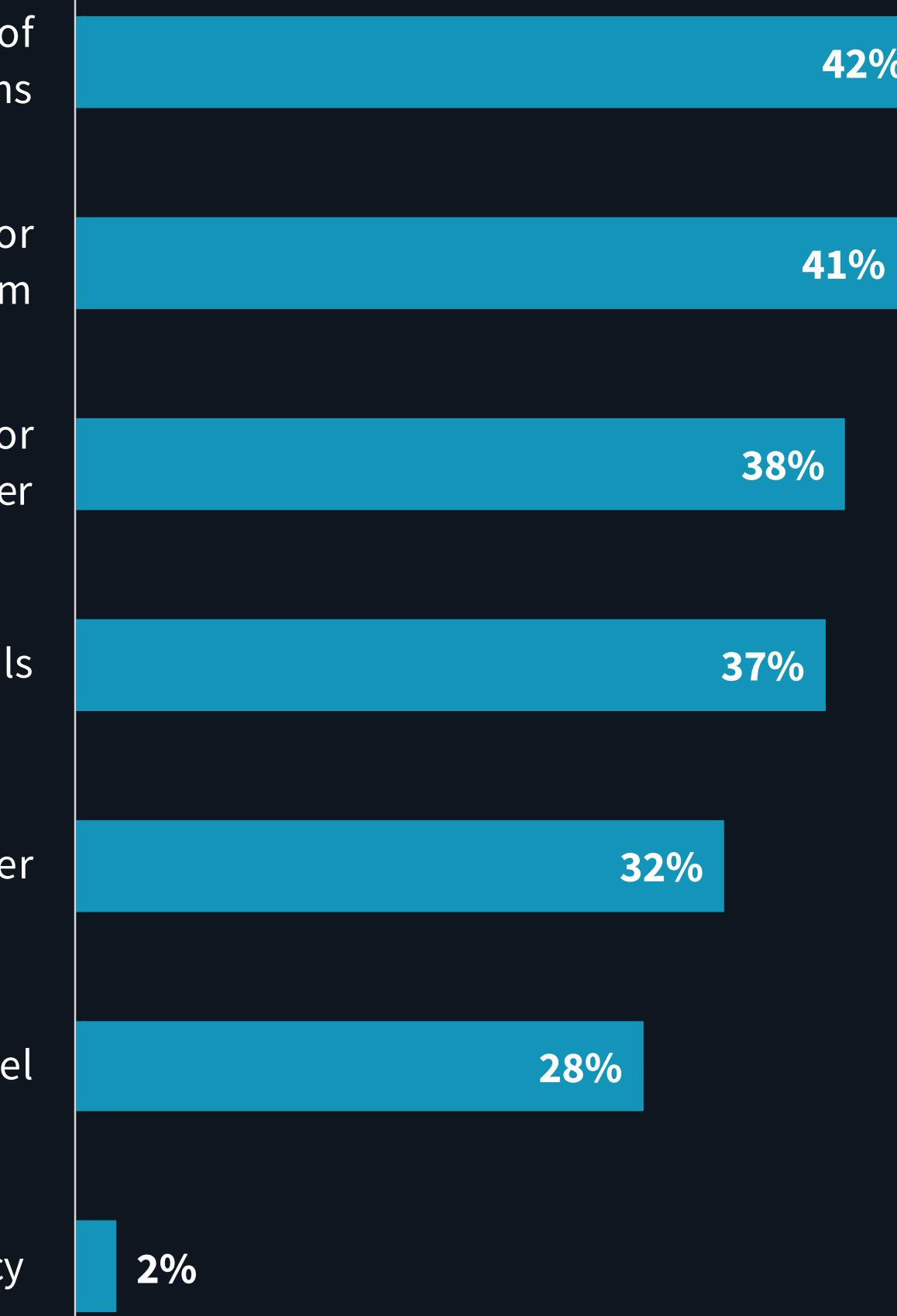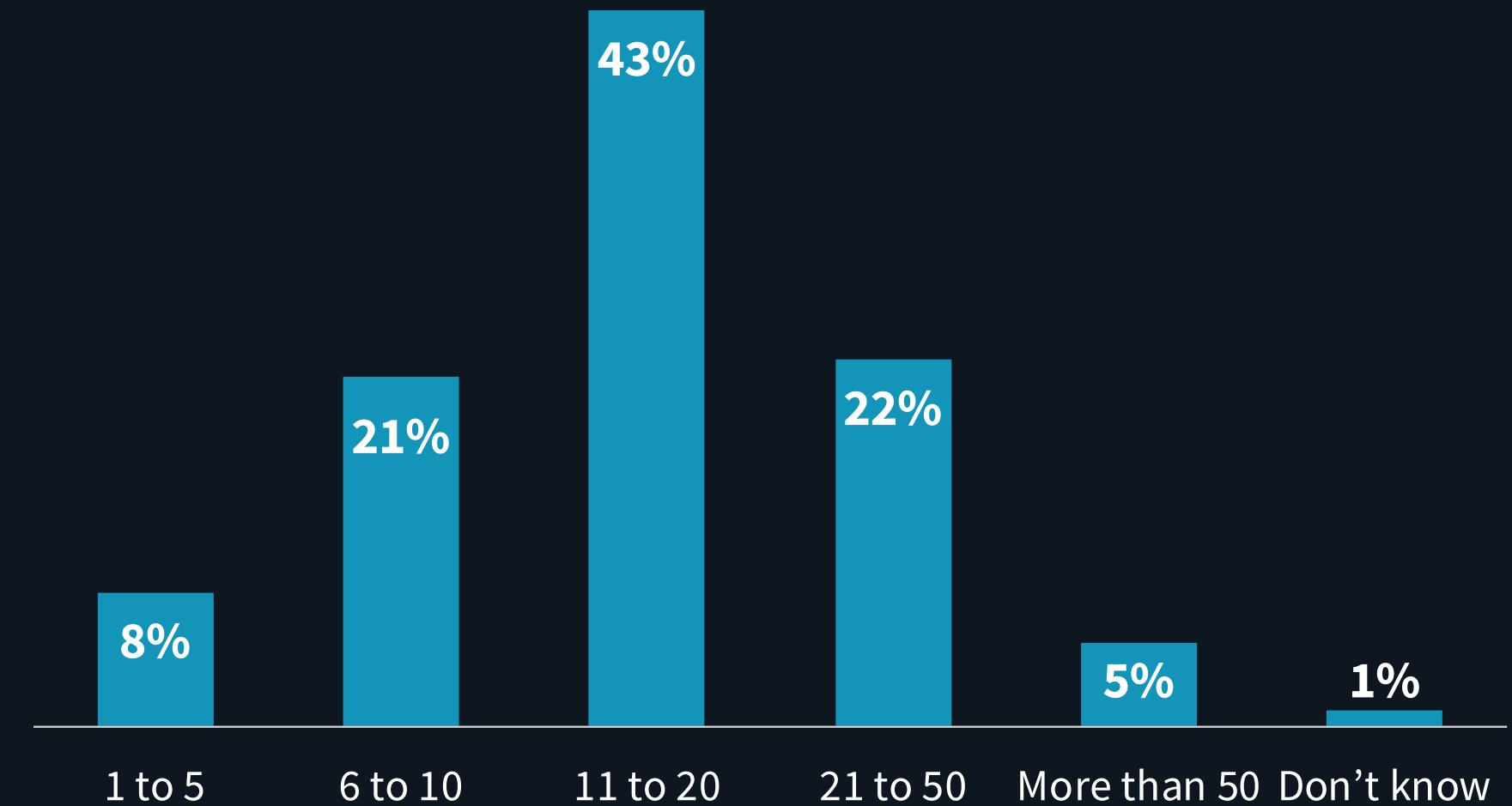
The proliferation of AppSec testing tools is an issue for many, with more than a third focusing investments on consolidation.
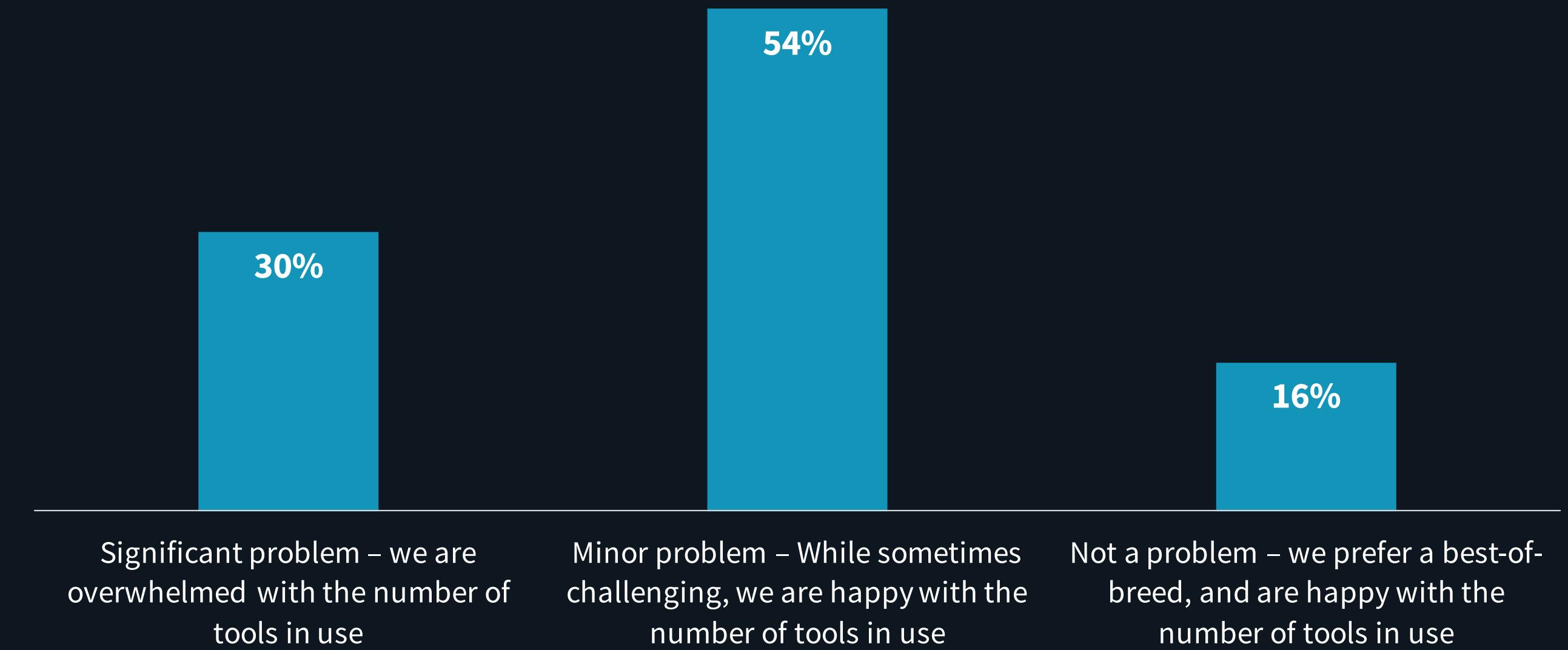
AppSec Tools Proliferation Is Driving Investments to Consolidate

Like other categories of security controls, many organizations are employing so many tools that they are struggling to integrate and manage them. This all too often results in a reduction in the effectiveness of the program and directs an inordinate amount of resources to managing tools. With 72% utilizing more than ten tools, complexity is becoming a key issue.

Individual AppSec tools in use

1 to 5	6 to 10	11 to 20	21 to 50	More than 50	Don't know
8%	21%	43%	22%	5%	1%

Tools proliferation is a problem for many
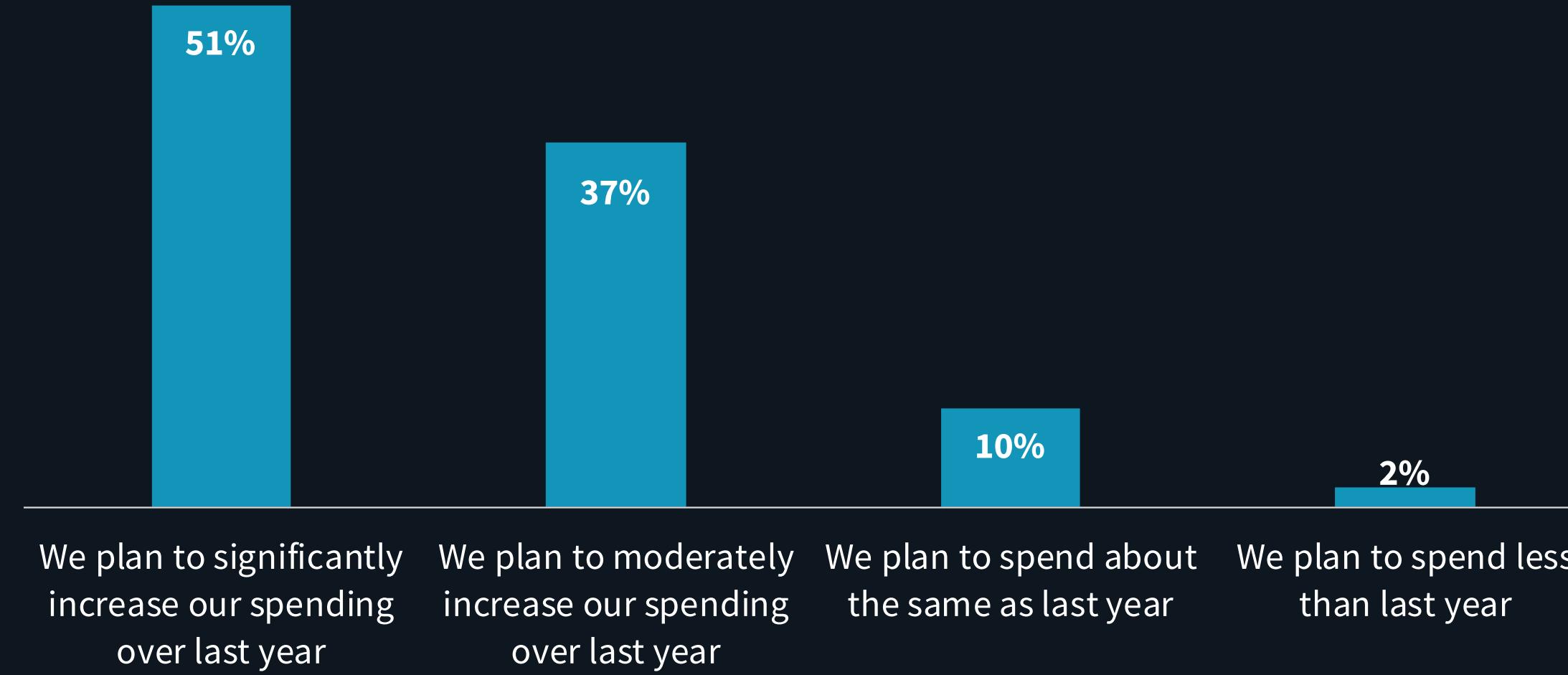
Significant problem – we are overwhelmed with the number of tools in use	Minor problem – While sometimes challenging, we are happy with the number of tools in use	Not a problem – we prefer a best-of-breed, and are happy with the number of tools in use
30%	54%	16%

Organizations are investing, with more than half planning to significantly increase spending on application security over the prior year.
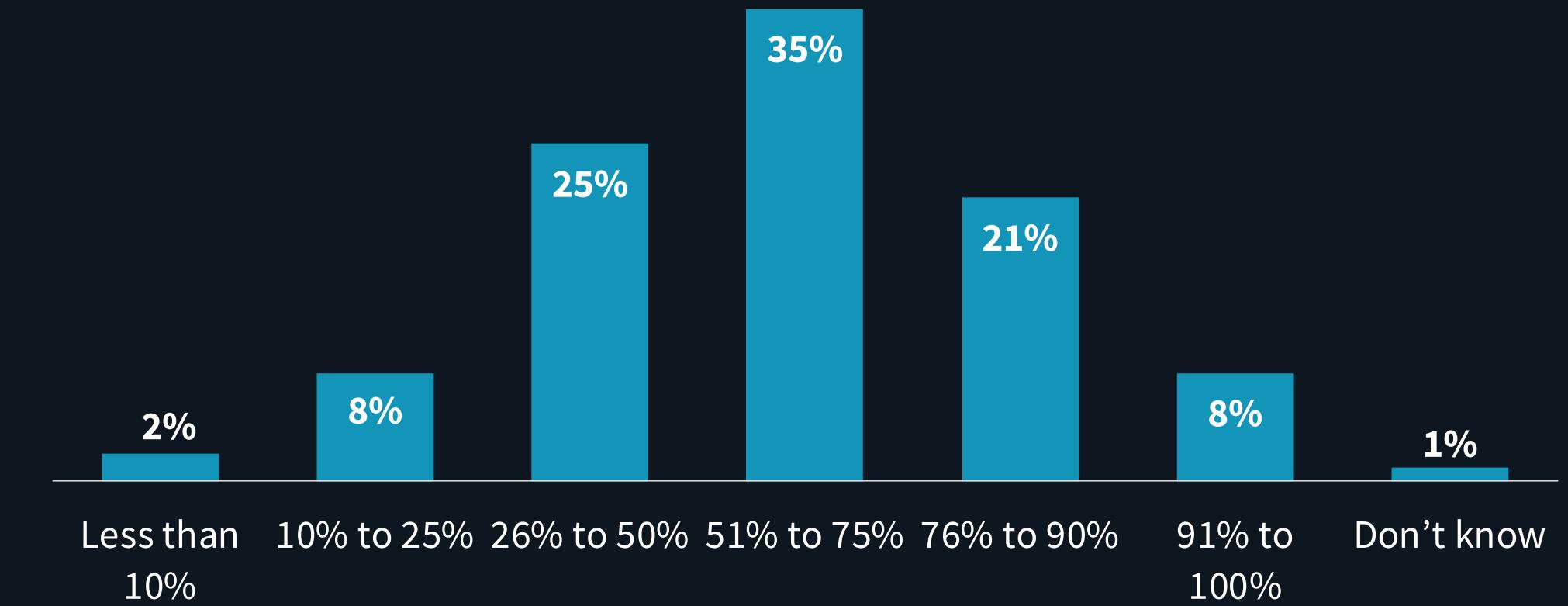
AppSec Spending Will Continue, but Code Coverage Will Still Be Lacking

While most expect to significantly increase application security spending over last year, only 30% will protect more than three-quarters of their codebase 12 months from now.

**Expected budget change for AppSec tools over the next 12 months**

- 51% — We plan to significantly increase our spending over last year
- 37% — We plan to moderately increase our spending over last year
- 10% — We plan to spend about the same as last year
- 2% — We plan to spend less than last year

Percentage of codebase expected to be protected by AppSec tools within 12 months

- 2% — Less than 10%
- 8% — 10% to 25%
- 25% — 26% to 50%
- 35% — 51% to 75%
- 21% — 76% to 90%
- 8% — 91% to 100%
- 1% — Don't know

AppSec Tools Investments Vary, Leaning Toward Securing Cloud Application Development

While 44% plan on targeting application security investments toward cloud, one-third are focused on consolidation of tools to simplify the process. Others plan to invest in expanding the use of testing tools to a higher percentage of their development teams and applications.

Application security investment priorities for next 12 months

1%
1%
4%
17%
43%
34%

- Our investments are weighted toward securing our cloud application development process
- Our investments are focused on consolidation of tools to simplify the overall process
- Our investments are focused on deploying application security across a higher percentage of our development teams and applications
- Our investments are focused on improving the efficacy of our application security program
- None of the above
- Don't know

VERAC○1DE

Veracode is the leading independent AppSec partner for creating secure software, reducing the risk of security breach, and increasing security and development teams' productivity. As a result, companies using Veracode can move their business, and the world, forward. With its combination of process automation, integrations, speed, and responsiveness, Veracode helps companies get accurate and reliable results to focus their efforts on fixing, not just finding, potential vulnerabilities.

Veracode serves more than 2,500 customers worldwide across a wide range of industries. The Veracode solution has assessed more than 19 trillion lines of code and helped companies fix more than 52 million security flaws.

Learn more at www.veracode.com, on the Veracode blog and on Twitter.

LEARN MORE

About ESG

Enterprise Strategy Group is an IT analyst, research, validation, and strategy firm that provides market intelligence and actionable insight to the global IT community.
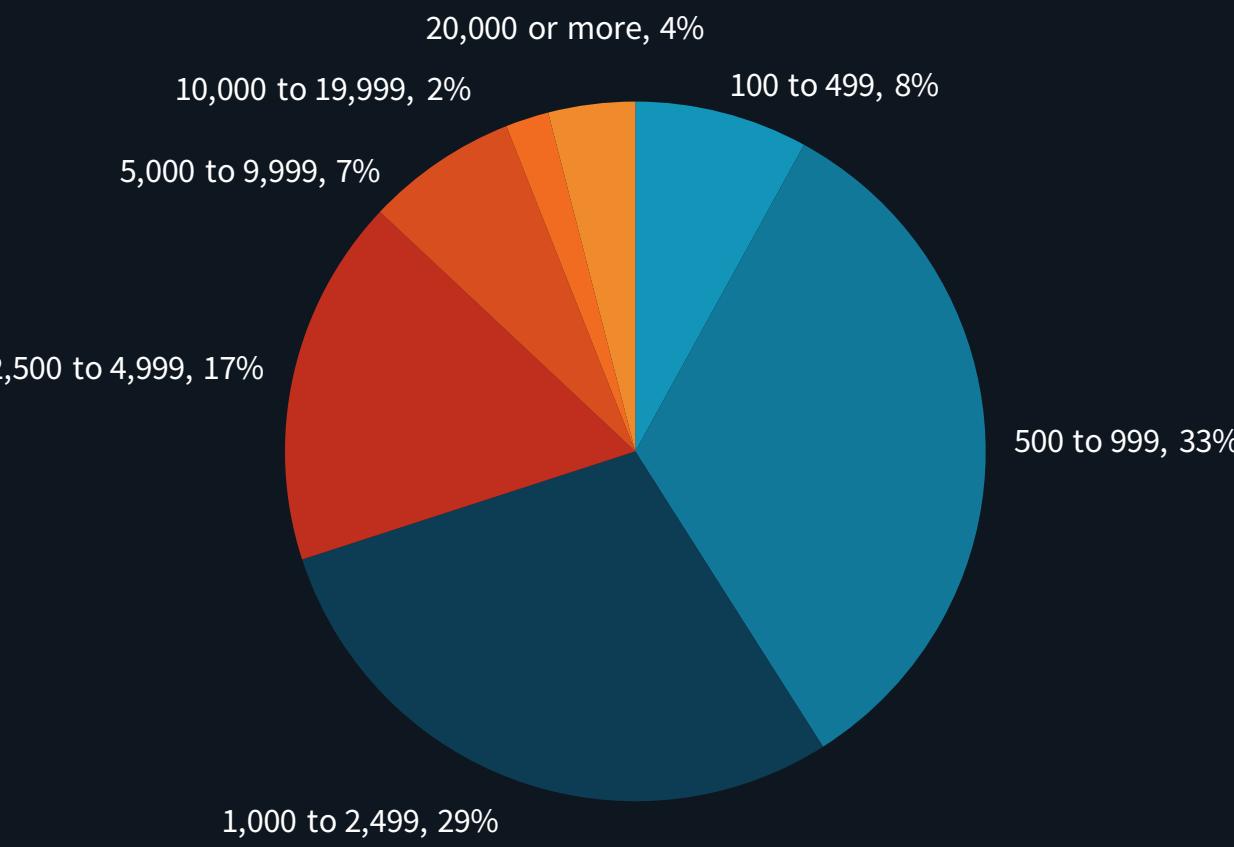
Research Methodology
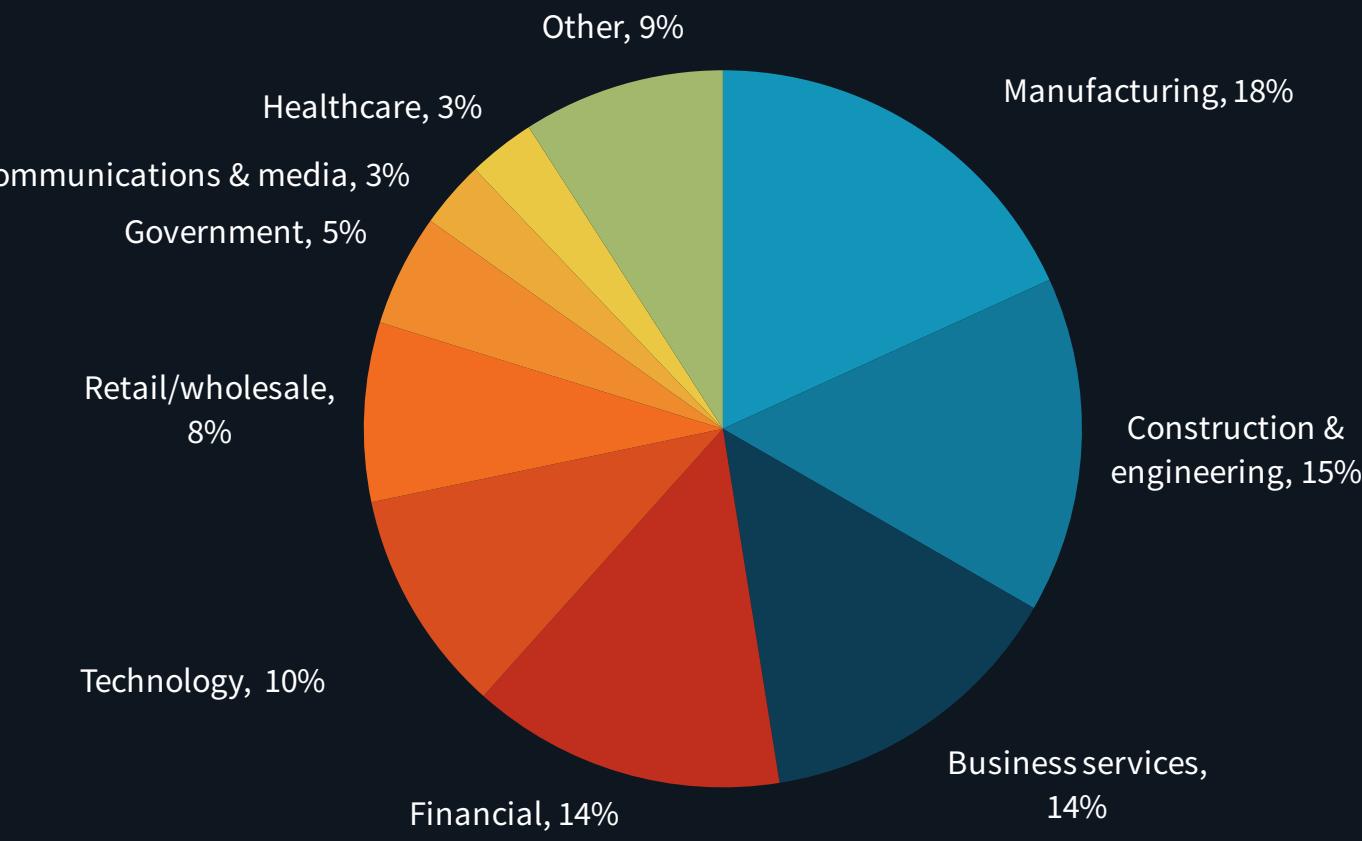
To gather data for this report, ESG conducted a comprehensive online survey of IT and cybersecurity professionals from private- and public-sector organizations in North America (United States and Canada) between June 12, 2020 and June 20, 2020. To qualify for this survey, respondents were required to be IT or cybersecurity professionals with insight and responsibility for securing AppDev technologies and processes or application development professionals involved with securing application development tools and processes. All respondents were provided an incentive to complete the survey in the form of cash awards and/or cash equivalents.

After filtering out unqualified respondents, removing duplicate responses, and screening the remaining completed responses (on a number of criteria) for data integrity, we were left with a final total sample of 378 IT, cybersecurity, and application development professionals.
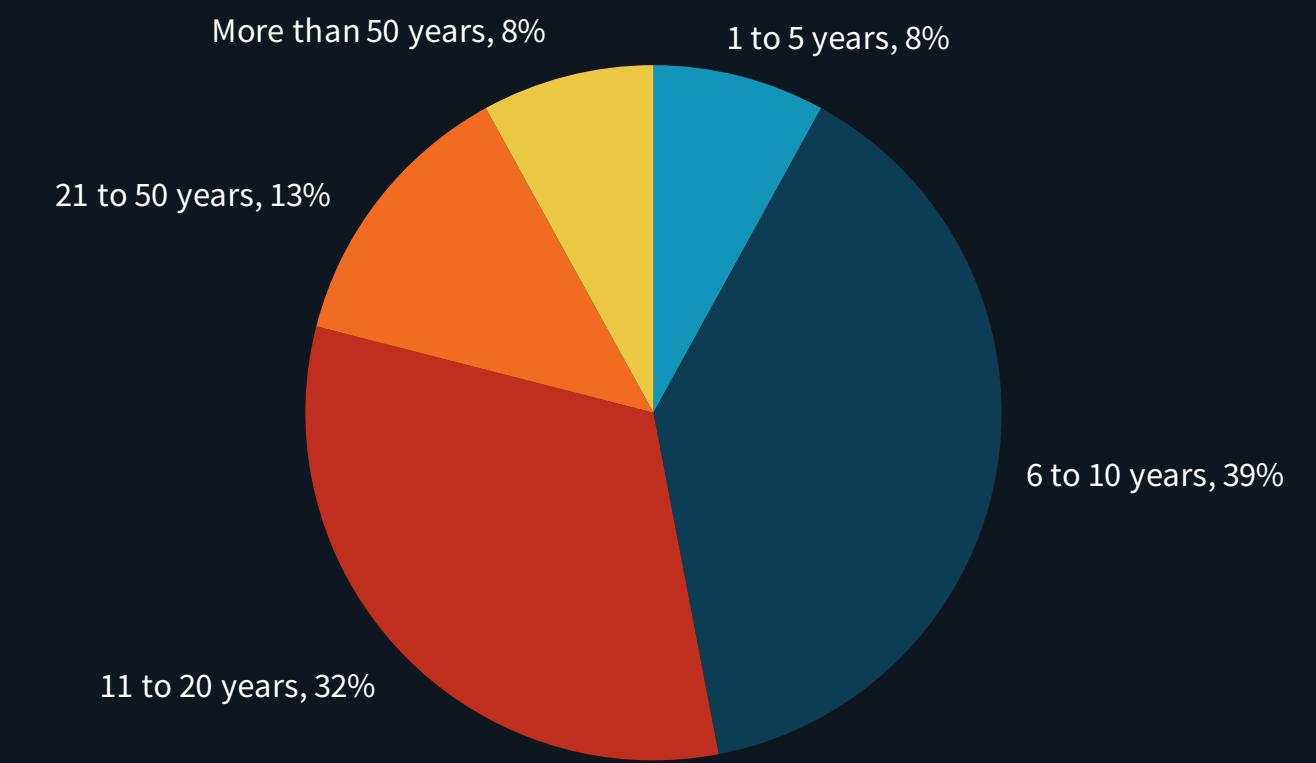
RESPONDENTS BY NUMBER OF EMPLOYEES

- 20,000 or more, 4%
- 10,000 to 19,999, 2%
- 5,000 to 9,999, 7%
- 2,500 to 4,999, 17%
- 1,000 to 2,499, 29%
- 100 to 499, 8%
- 500 to 999, 33%

RESPONDENTS BY INDUSTRY

- Other, 9%
- Healthcare, 3%
- Communications & media, 3%
- Government, 5%
- Retail/wholesale, 8%
- Technology, 10%
- Financial, 14%
- Manufacturing, 18%
- Construction & engineering, 15%
- Business services, 14%

RESPONDENTS BY AGE OF ORGANIZATION

- More than 50 years, 8%
- 1 to 5 years, 8%
- 21 to 50 years, 13%
- 6 to 10 years, 39%
- 11 to 20 years, 32%

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