digicert®

The State of Software Supply Chain Security

Research Report | 2026

Abstract

Virtual Intelligence Briefing (ViB) conducted a survey of IT and security professionals who are responsible for software supply chain security.

A majority overstate the maturity of their supply chain security programs, with confidence outpacing actual rates of automation, compliance readiness, and SBOM/code signing. These gaps can be closed, however, by focusing on automation of signing and security checks, defining and enforcing policies, deploying the right tools, and more thoroughly embedding security into software development workflows.

Introduction

This report is based on a survey of IT and security professionals responsible for software supply chain security.

The survey reveals a cohort that overestimates the maturity of their software supply chain security efforts even as they grapple with significant challenges.

Confidence in supply chain security is surprisingly high, while execution lags behind. A relatively small portion of respondent organizations use automation, create SBOMs, or sign code. Compliance readiness is low.

Nearly 7 in 10 are behind in their preparation for post-quantum cryptography (PQC).

It is possible to close these gaps, however, by focusing on automation and policy, deploying the right tools, and more thoroughly embedding security into software development workflows.

Demographic Summary

This report represents the views of 222 IT and security professionals.

They work mostly at mid-sized organizations, with 31% at companies with between 1,000 and 2,499 employees and 36% with over 2,500.

They come from a variety of sectors, the three most highly represented being:

20%	14%	11%	11%
Software	Business Services	Education	Heathcare

35% work at the Director level or higher.

Given the ability to select more than one role, 60% said they work in cybersecurity functions, 55% said infrastructure/IT Ops, and 54% selected IT architecture.

For a full breakdown of respondent demographics, see the Appendix.

Key Takeaways

The data suggests a maturity paradox—while almost half of organizations rate the maturity of their supply chain security program maturity as “established” or “optimizing,” many indicators of maturity are lagging, e.g., only 13% fully automate code signing.

The maturity paradox appears to be due to a lack of preparation for supply chain security and execution, including:

55% A lack of internal expertise

55% Budget constraints

41% Development team resistance

34% A lack of centralized visibility across all security tools

71% Partial automation of security checks or ad-hoc automation

---

Small percentages of organizations embody the characteristics of a mature supply chain security program:

11%
of respondents are actively providing SBOMs today, with accuracy and integration difficulties getting in the way of SBOM implementation

12% say they “always” sign container images

10% are testing or implementing PQC for code signing, despite looming deadlines

17% always sign their SBOMs

12% of respondents report being fully prepared to meet regulatory requirements

---

Organizations with mature software supply chain security tend to:

- Formalize policies before scaling automation
- Automate comprehensively
- Embed security into development workflows
- Secure key storage
- Prepare for tomorrow’s requirements

Tools, Practices, and Platforms

Respondents use a wide variety of tools and platforms for software supply chain security. For full details, see the Appendix.

Highlights:

Tools used for code signing

35% use a cloud provider's code signing services, e.g., AWS Signer

33% use DigiCert Software Trust Manager or KeyLocker

Code signing certificate management

46% use DigiCert for code signing certificate management, e.g., CertCentral

34% use a cloud provider's native tools, e.g., AWS KMS

Container/Kubernetes deployment rate

26% of respondents deploy over 51% of their applications on containers

59% deploy over 25% of their applications on containers

CI/CD platforms used

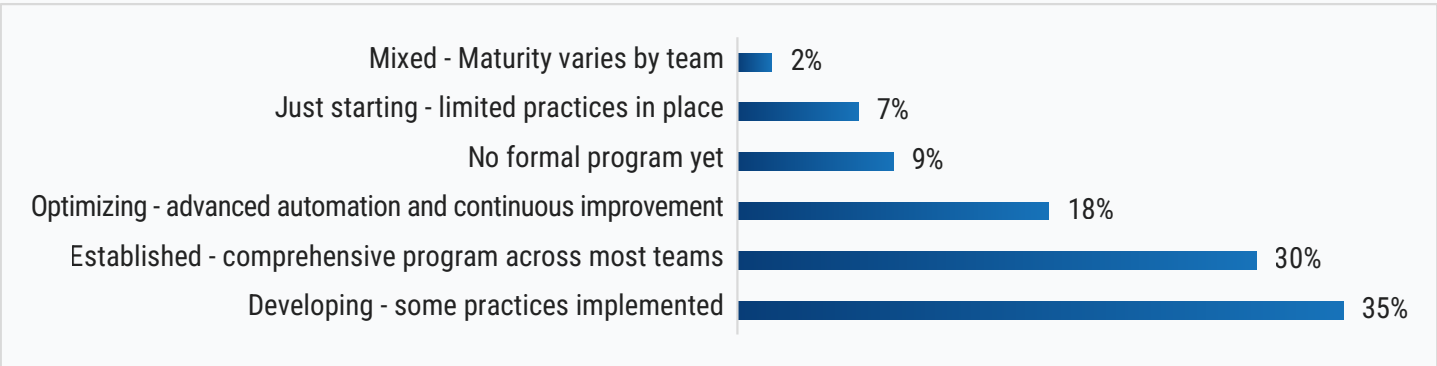
55% use Azure DevOps

40% use GitHub Actions

The Maturity Paradox

The survey reveals an apparent paradox regarding perceptions of software supply chain security. Nearly half of organizations consider their supply chain program maturity to be either "established" or "optimizing." Overall, 83% have established a comprehensive program or are developing one. Just 9% said they have no formal program in place.

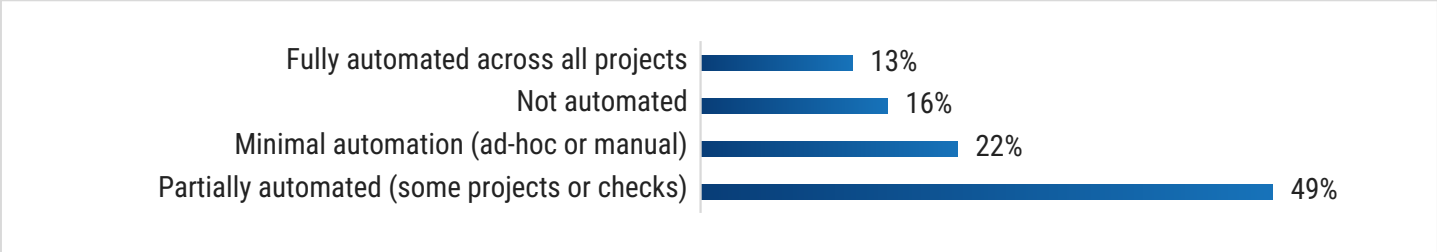
Perceptions of supply chain security program maturity



How mature is your organization's software supply chain security program?

These numbers would suggest the presence of robust, mature programs for supply chain security. Maturity seems to lag, however, when organizations are measured by key indicators of supply chain security maturity. These include rates of automation, automated code signing, and software bill of materials (SBOM) creation. For instance, only 13% fully automate code signing, and 13% fully automate security checks.

Level of automation for security checks



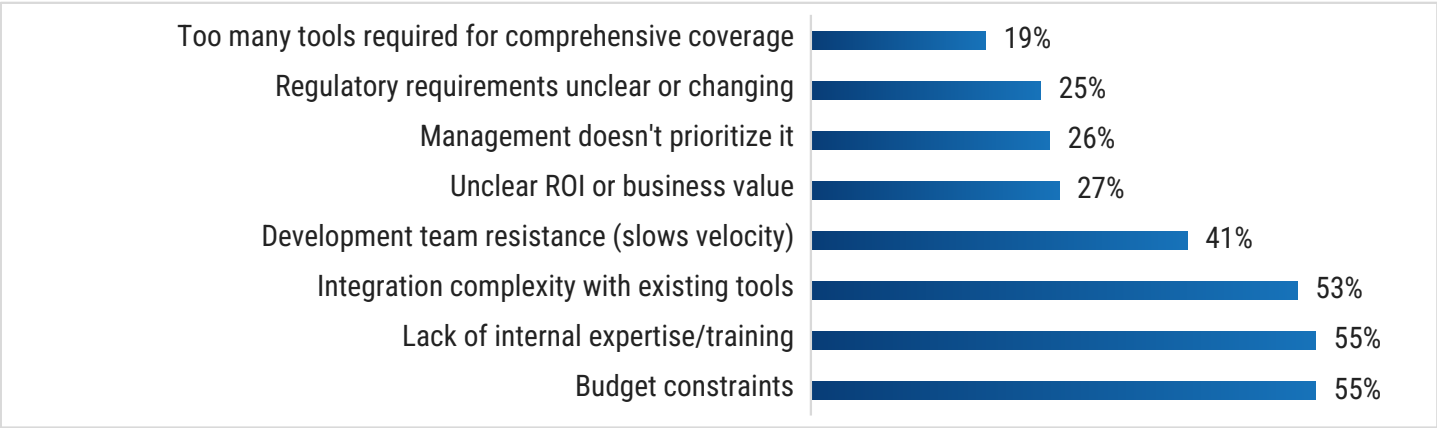
Do you currently automate security checks (SAST, DAST, SCA, vulnerability scanning) in your pipelines?

The data also shows a confidence gap: 19% of those with established programs lack confidence in audits. A mature program should give stakeholders confidence that they can pass an audit. For C-Level executives, this number jumps to 40%.

Unpacking the Maturity Paradox: The Limiting Role of Preparation and Execution

What's behind the maturity paradox? A deeper look reveals obstacles to preparation and execution. For example, when asked to select challenges preventing faster adoption of supply chain security practices, 55% said "budget constraints." The result was similar when the survey asked the question from a different direction: "What would most improve your current software supply chain security posture?" 47% of respondents said that more budget/resources would help them improve.

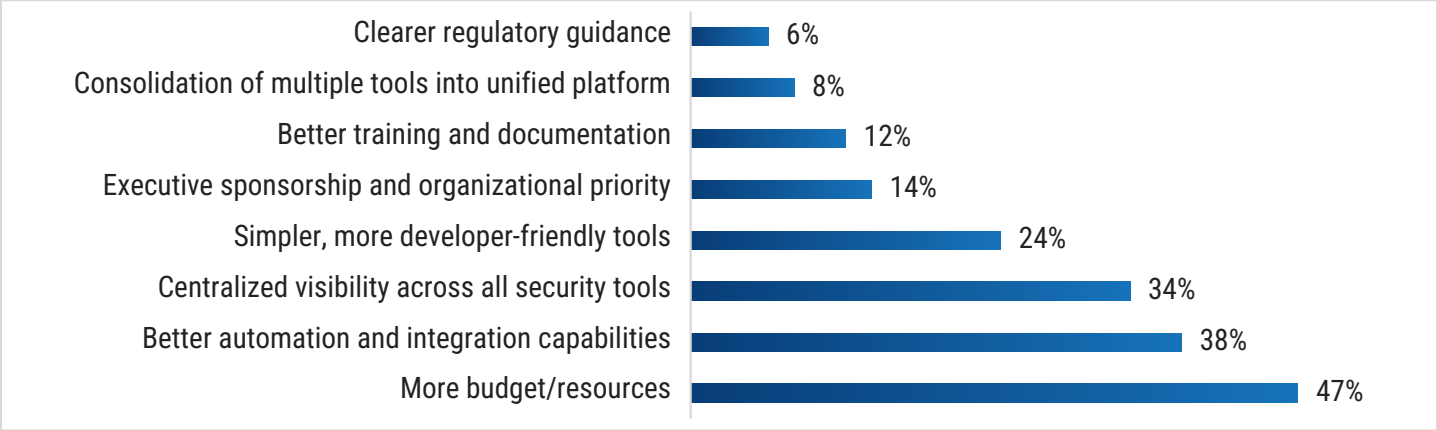
Challenges preventing faster adoption of supply chain security practices



Which of these challenges are preventing faster adoption of supply chain security practices?

A lack of internal expertise and training emerged as a challenge, too, cited by 55%. This issue may be related to budget, but a skills gap is probably more than just a money problem. The rapid evolution of supply chain security practices is likely outstripping workforce development. Further to this idea, some challenges are cultural in nature: 41% said development team resistance was slowing the adoption of supply chain security practices.

Factors that would improve supply chain security posture



What would most improve your current software supply chain security posture?

Integration emerged as a challenge to adoption and a possible issue to resolve in improving supply chain security. 53% said that integration complexity for existing tools created challenges for supply chain security adaptation, while 38% said better automation and integration capabilities would lead to improvements.

SBOM requirements outpace capabilities

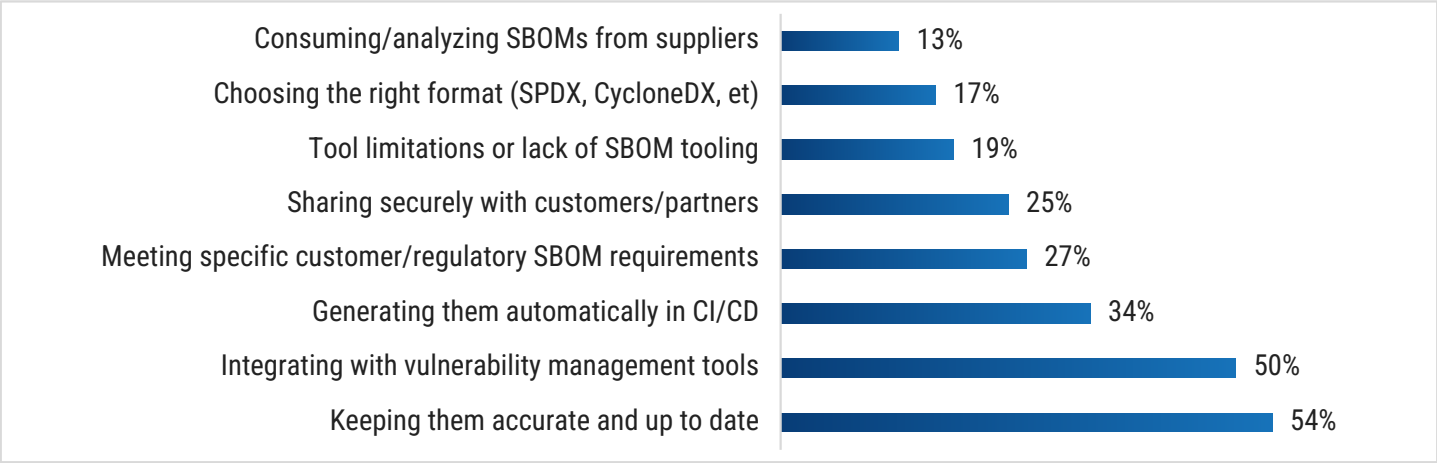
The percentage of software customers expected to require SBOMs is likely to increase in the coming years. More and more enterprises and OEM partners want to know exactly what's inside the code they're using. In this context, it is striking that just 11% of respondents are actively providing SBOMs today. Just 27% said they have no SBOM requirement today and do not expect to have one in the future. 43% are expecting to face an SBOM requirement in the next 24 months, while 19% have an SBOM requirement, but are still implementing the SBOM process.

Requirement to provide SBOMs

Are you currently required by customers or regulators to provide an SBOM?

What's holding back SBOM implementation? Creating SBOMs is not a push-button process, and a number of challenges inhibit adoption. These include concerns about accuracy (cited by 54%) and difficulties integrating with vulnerability management tools (50%). Automation, which many view as essential for SBOM success, is seen as a challenge by 34%.

SBOM challenges

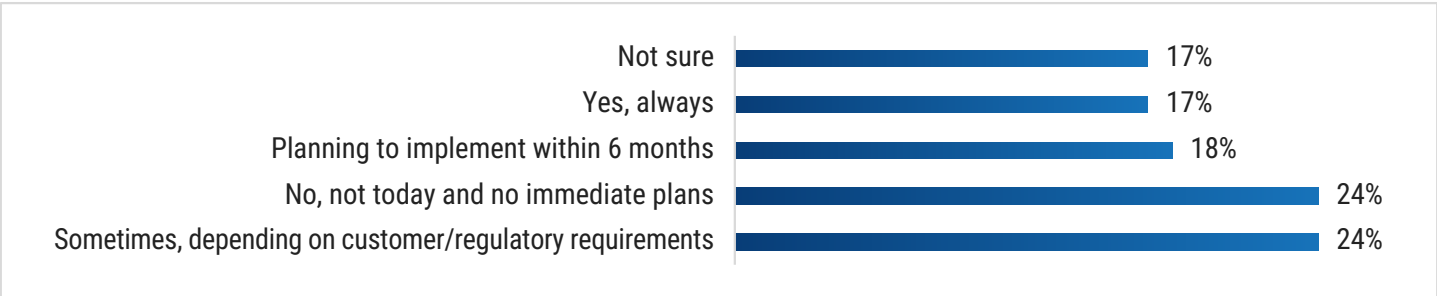


What are your biggest challenges with SBOMs today?

Inconsistent rate of signing SBOMs

SBOM signing, which creates the digital equivalent of a tamper-proof seal on the SBOM, proves the authenticity and integrity of the document. For this reason, a signed SBOM is an indicator of supply chain security maturity. Respondents report inconsistent rates of SBOM signing, however. Only 17% always sign their SBOMs. A quarter sign sometimes, depending on requirements, while 18% plan to implement signing within six months.

Frequency of SBOM signing

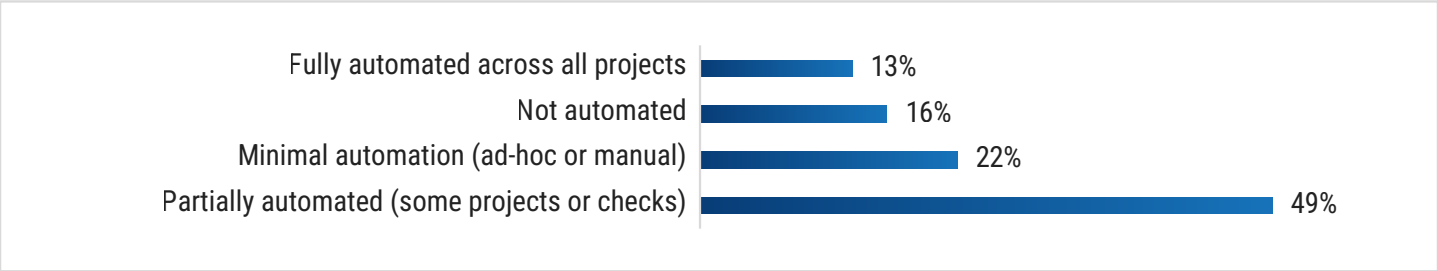


Do you currently sign your SBOMs to ensure integrity?

The automation gap for security checks

Automating security checks like static application security testing (SAST) and dynamic application security testing (DAST) makes supply chain security more efficient and accurate. The more fully automated an organization's security check process, the more mature its supply chain security program will be. Yet, only 13% are fully automated, while 71% are either partially automated or automated on an ad-hoc basis. Considering factors like complexity and integration difficulties, it's understandable that not all organizations are fully automated.

Automation of security checks

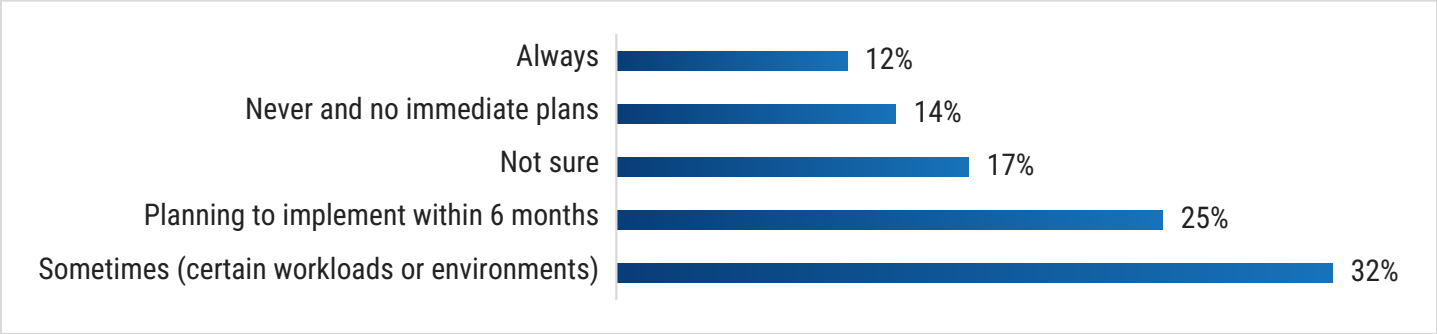


Do you currently automate security checks (SAST, DAST, SCA, vulnerability scanning) in your pipelines?

Container security lags

Container use is popular, with 59% of respondents saying they deploy over 25% of their applications in containers like Kubernetes. But container signing, a step that ensures authenticity and prevents supply chain attacks, is less common. Only 12% say they "always" sign container images, indicating supply chain security maturity, while 32% do it "sometimes." One quarter plan to start signing containers within six months.

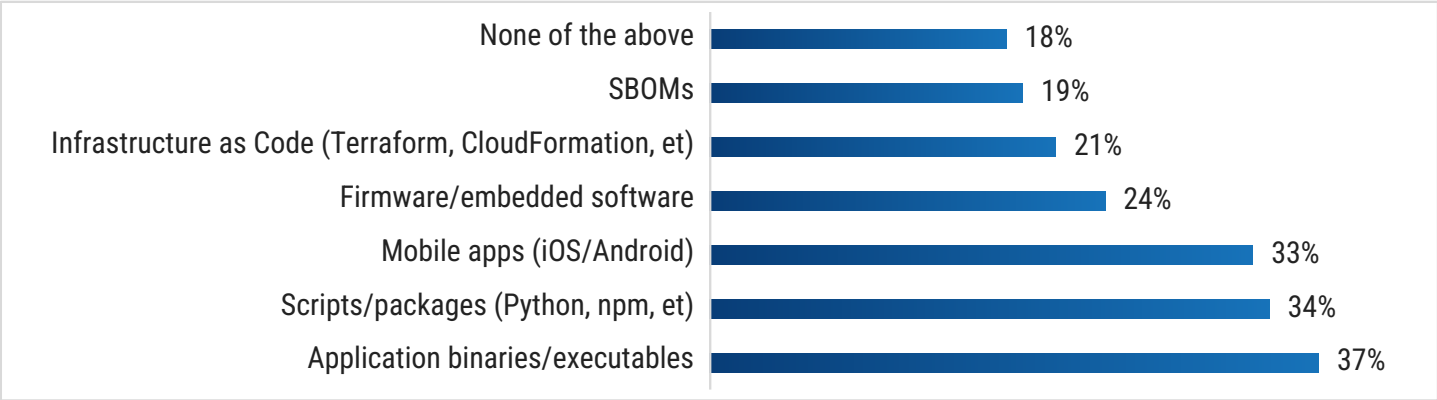
Frequency of signing of container images



Do you currently sign container images before deploying to production?

Respondents also sign a variety of other types of software and artifacts. These include application binaries and executables (signed by 37%) and scripts, like Python and npm (34%). These findings suggest that respondents view signing as an important security control to mitigate supply chain risk.

Other software/artifacts signed



Beyond containers, which other types of software/artifacts do you sign?

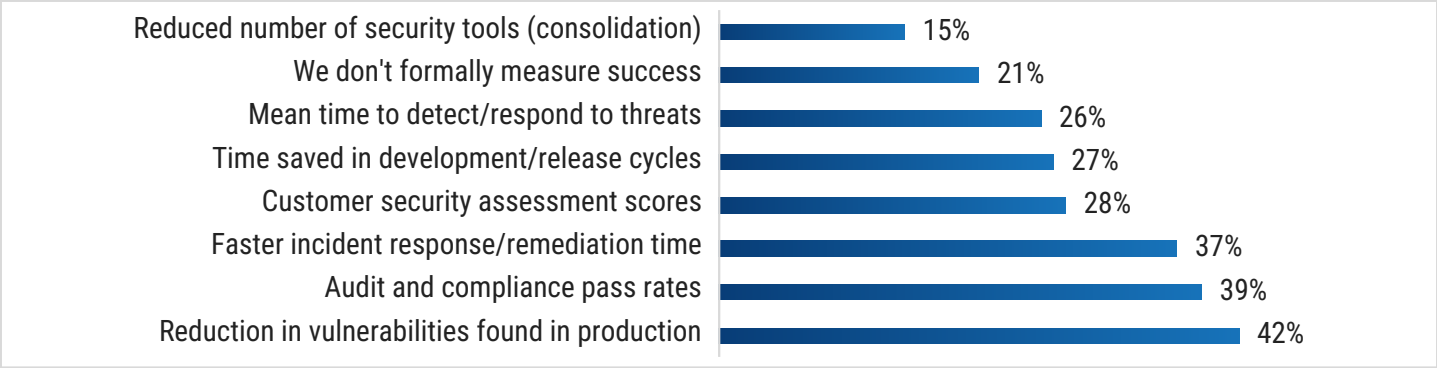
A focus on measuring lagging indicators of success

One-fifth of organizations do not formally measure the success of their supply chain security initiatives.

This is a high number, considering the threat environment they're working in, along with customer expectations and regulatory and audit requirements. The most popular measurements are lagging indicators such as vulnerability reduction (41%) and audit and compliance pass rates (38%).

As organizations evaluate new ways of measuring how they're doing with supply chain security, they should consider more forward-looking factors like shift-left metrics and rates of automation coverage.

Measuring the success of supply chain security initiatives

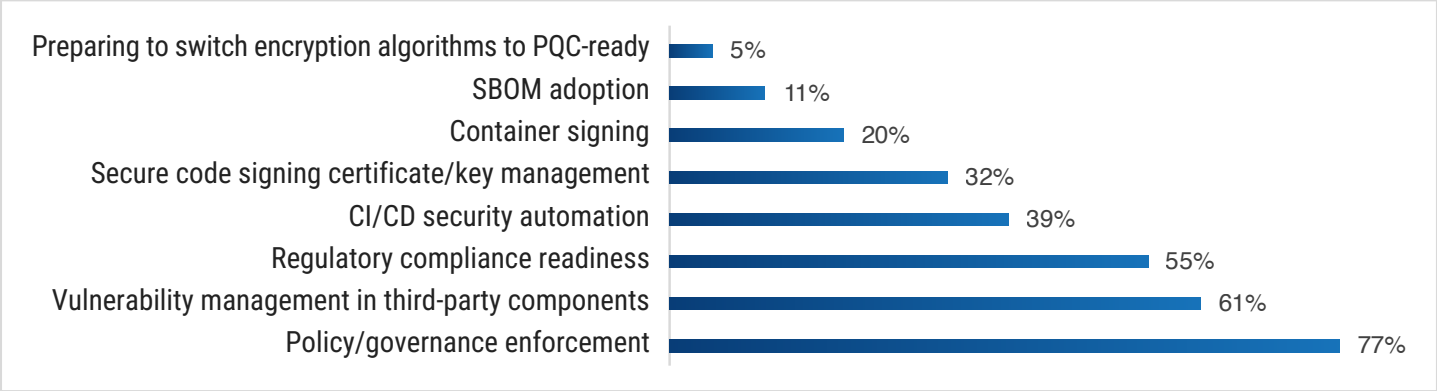


How does your organization measure the success of software supply chain security initiatives?

How the maturity paradox shows up in supply chain security priorities

The maturity paradox also surfaces in top priorities for the next 12 to 18 months. The most common priority is policy and governance enforcement, selected by 77%, followed by vulnerability management in third-party components (61%), and regulatory compliance readiness (55%). While all three are worthy goals, they do not bear much on supply chain security maturity. CI/CD security automation (39%), secure code signing (32%), and container signing (20%), all required for maturity, appear to be lower priorities.

Top priorities for the next 12-18 months

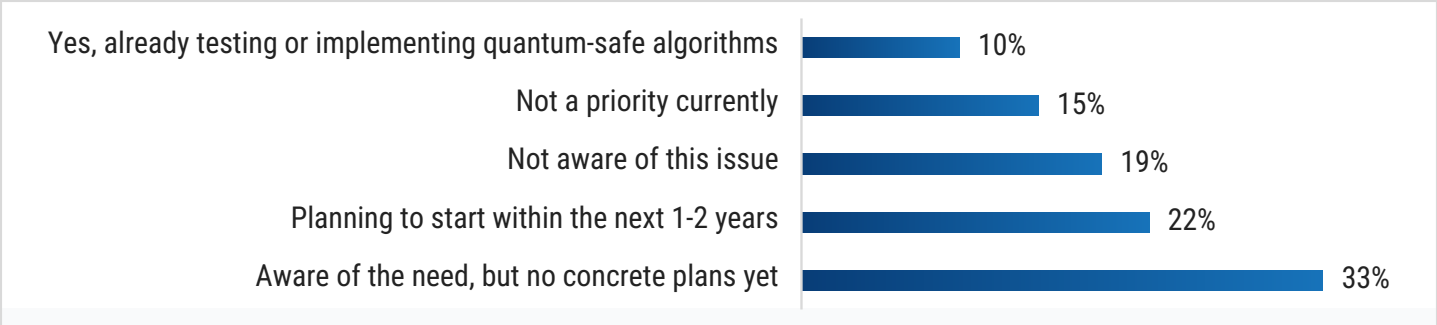


Which are your top priorities for the next 12-18 months?

The Concerning Lack of Priority for Post-Quantum Cryptography

Obstacles to supply chain security preparation and execution also appear in a concerning lack of priority for post-quantum cryptography (PQC). The EU and US federal government mandate PQC adoption starting in 2030, so the transition should now be underway. Despite this looming deadline, 68% of respondents are unaware of the issue, have no plans, or are not prioritizing post-quantum cryptography for code signing. Only 10% are testing or implementing PQC, though 33% are aware of the need and 22% plan to start in the next two to three years.

State of preparation for PQC



Has your organization begun preparing for post-quantum cryptography for code signing?

It would appear that all but a small number of organizations are going to miss the deadline. This portends:

- ⊗ Compressed adoption timeline
- ⊗ Compliance gaps
- ⊗ Quantum breach risk exposure

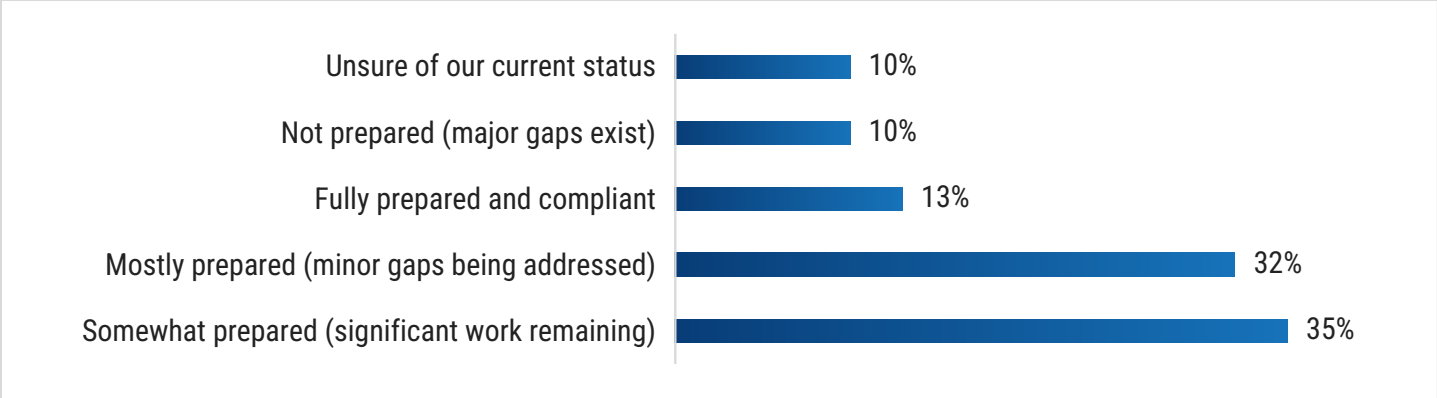
The Regulatory Compliance Preparedness Gap

Compliance is another area where supply chain security preparation and execution also appear to lag. The major regulatory frameworks mandate software supply chain security controls. Compliance affects the organizations surveyed in this study:

40% of respondents try to comply with ISO/IEC 27001, while 31% must comply with PCI DSS standards. A quarter want to comply with the NIST Secure Software Development Framework (SSDF). The US government's Executive Order 14028 is relevant to 14%.

In this context, it is notable that only 12% of respondents report being fully prepared to meet regulatory requirements. 55% have preparation work to do, in some cases quite extensive.

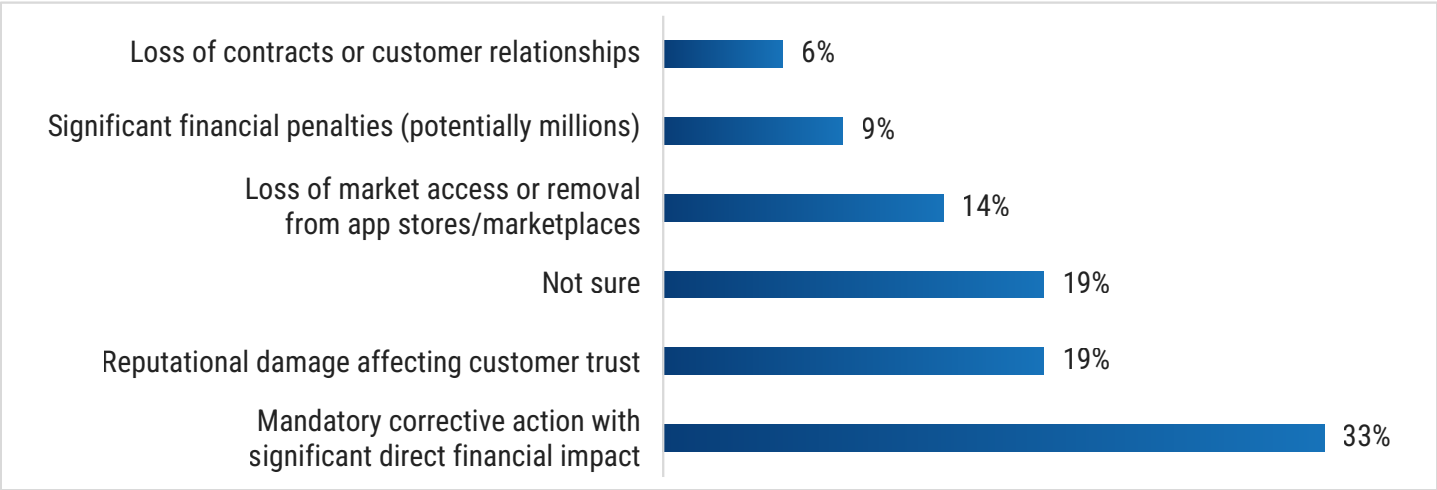
Levels of preparation to meet regulatory requirements



How prepared is your organization to meet these regulatory requirements?

Respondents are at risk of failing compliance audits related to software supply chain security. Asked about the biggest impacts of such a failure, 35% said they would face mandated corrective action with significant financial impact. Another 19% said a failed audit would damage their reputations and customer trust.

Failing compliance audit impacts



What would be the biggest impact of failing a compliance audit?

What Defines Leaders vs. Laggards?

A number of patterns characterize organizations with mature software supply chain security.

The following five factors differentiate such leaders from laggards:

1. **Comprehensive automation:** Leaders have fully automated code signing and security checks, versus doing it selectively. A quarter of leaders are fully automated, compared to 5% of laggards. Leaders appear to recognize that partial automation creates risk asymmetry.

2. **Policy formalized before the scaling of automation:** It is not optimal to automate inconsistent or poorly defined policies. Compared to laggards, a higher percentage of leaders firm up their policies before moving on to automation.

3. **Preparation for tomorrow's requirements:** Among leaders, 23% are now testing PQC algorithms. 46% of large enterprises (2,500+ employees) are actively preparing for shorter certificate validity periods. Leaders don't wait for mandates.

4. **Secure key storage:** Leaders are more likely to store keys using secret vaults, cloud key management systems (KMS), or hardware security modules (HMS), whereas laggards tend to store keys on shared drives or local machines.

5. **Security embedded in development workflows:** Leaders are more likely to automate CI/CD, making it possible to secure the software supply chain without disrupting development workflows.

Strategic Recommendations

What should organizations do to become more effective in software supply chain security?

The following strategic recommendations emerge from the survey findings:

Close the automation gap before regulatory deadlines take effect
The automation of CI/CD processes, including signing, vulnerability scanning, and SBOM generation should be a priority for organizations facing the risk of noncompliance.

Establish formal code signing policies
With 89% of organizations lacking formal, enforced code signing policies, one developer mistake could result in the shipping of unsigned code the loss of keys. Formal code signing policies—defined and enforced—reduce the risk of this outcome.

Make SBOM signing a hard requirement
SBOMs without signatures are assertions without proof. However, only 17% of respondents always sign SBOMs, while others sign them sometimes. Such a conditional approach undermines trust. Instead, the best approach is to make SBOM signing a non-negotiable step in the software delivery process.

Start planning for PQC now
With 68% either unaware of post-quantum issues or lacking concrete plans, most organizations will face compressed migration timelines when the threat arrives (i.e., when a quantum computer becomes capable of attack). It's best to start now by inventorying use of cryptography, testing post-quantum algorithms in non-production environments, and developing migration roadmaps.

Consolidate tools to enable integration
The 38% believing that better automation would improve security and 34% seeking centralized visibility are both referencing the issue of tool sprawl. Too many tools usually mean incomplete integration and a lack of coherent workflows. Consolidating tools is a step toward better supply chain security and broader visibility.

Conclusion

The survey shows that software supply chain security is not where it needs to be, feelings of confidence aside.

The difficulties affecting organizations with immature chain security programs stem from deficits in preparation and execution. Automation levels are low, as are rates of code signing and SBOM signing. Audit readiness is weak, and too many organizations are putting off PQC despite the need to start the transition soon. These are solvable problems, though. The most mature organizations in the survey demonstrate the path to success.

With comprehensive automation, policy formalization, security embedded in development, and preparation for future requirements, it is possible to improve software supply chain security—reducing risk and negative security outcomes while bolstering levels of regulatory compliance.

About DigiCert

DigiCert is a leading global provider of digital trust and security solutions that acts as a Certificate Authority (CA) that issues digital certificates for identity verification and encryption, securing websites, software, devices (IoT), documents, and transactions.

The company offers enterprise-grade certificate management platforms like DigiCert® ONE, which automates trust and security across PKI, IoT, and signing workflows for major organizations worldwide.

DigiCert® Software Trust Manager, a cloud-based solution within the DigiCert ONE platform, has the potential to close the supply chain security gap and address many of the other challenges identified in this survey. Software Trust Manager makes this possible by managing code-signing certificates, keys, and workflows. It enables DevOps teams to automate signing, prevent tampering, and enforce compliance via FIPS-compliant HSMs.

To learn how DigiCert can help accelerate your maturity with Software Supply Chain Security, visit digicert.com

This vendor neutral, 3rd party research was independently conducted by ViB.

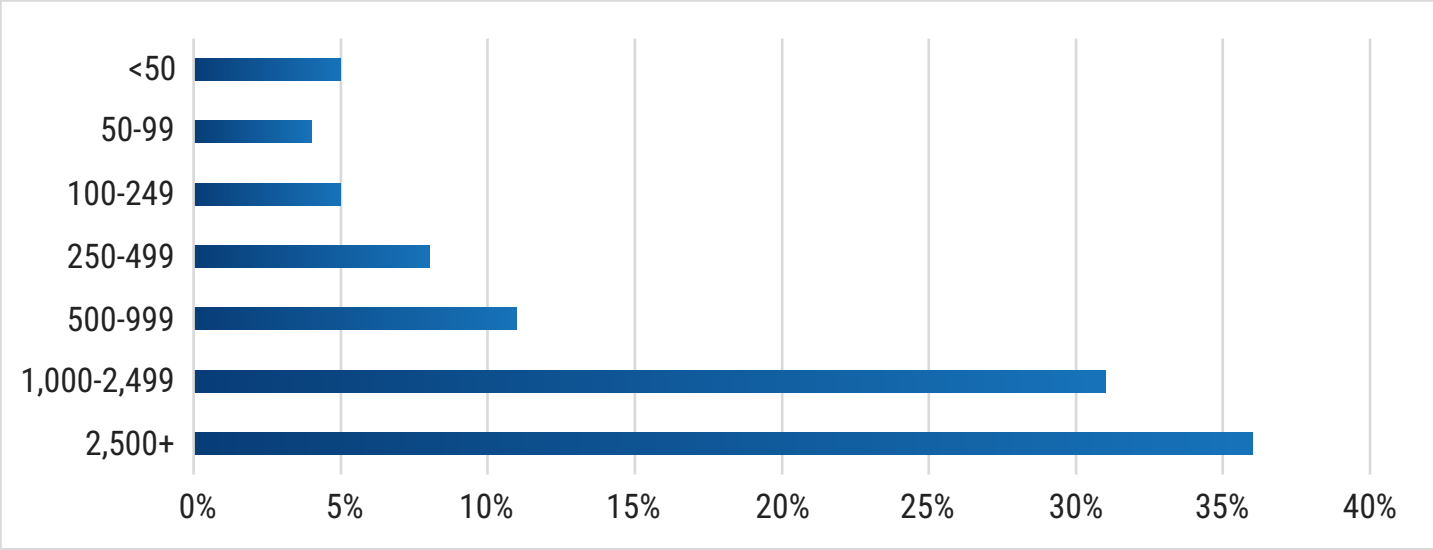
ViB's best-in-class market research design and analysis methodology delivers the industry's most accurate insights from precisely targeted, highly engaged members of ViB's 10M+ strong community of ViB's Technology Professionals.

ViB leverages the market research industry's best practices and tools, incorporating extensive quality controls across the entire lifecycle from survey design to analysis, and presentation of findings.

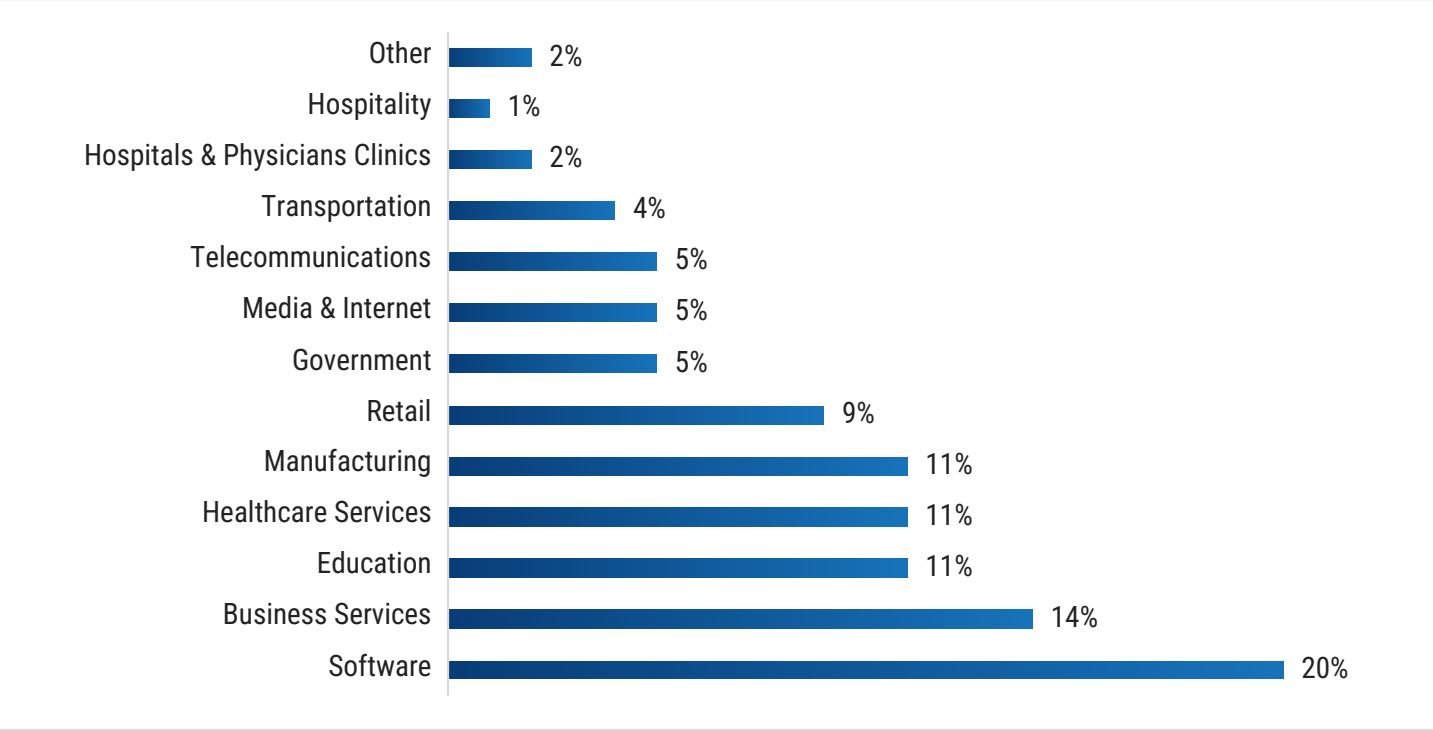
Appendices

Respondent Demographic Details

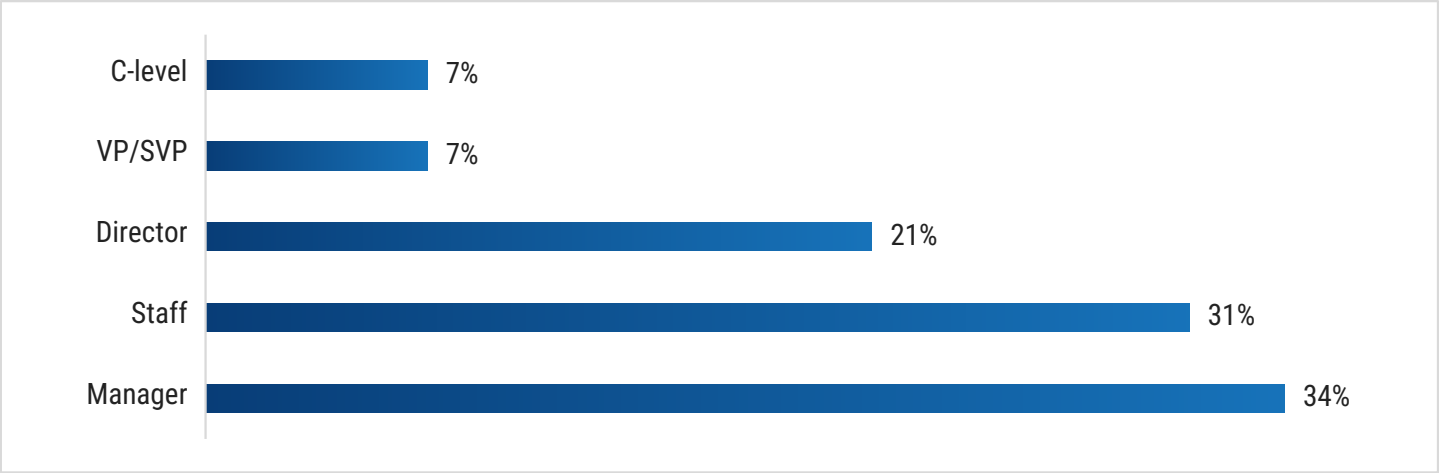
Size of organization (employees)



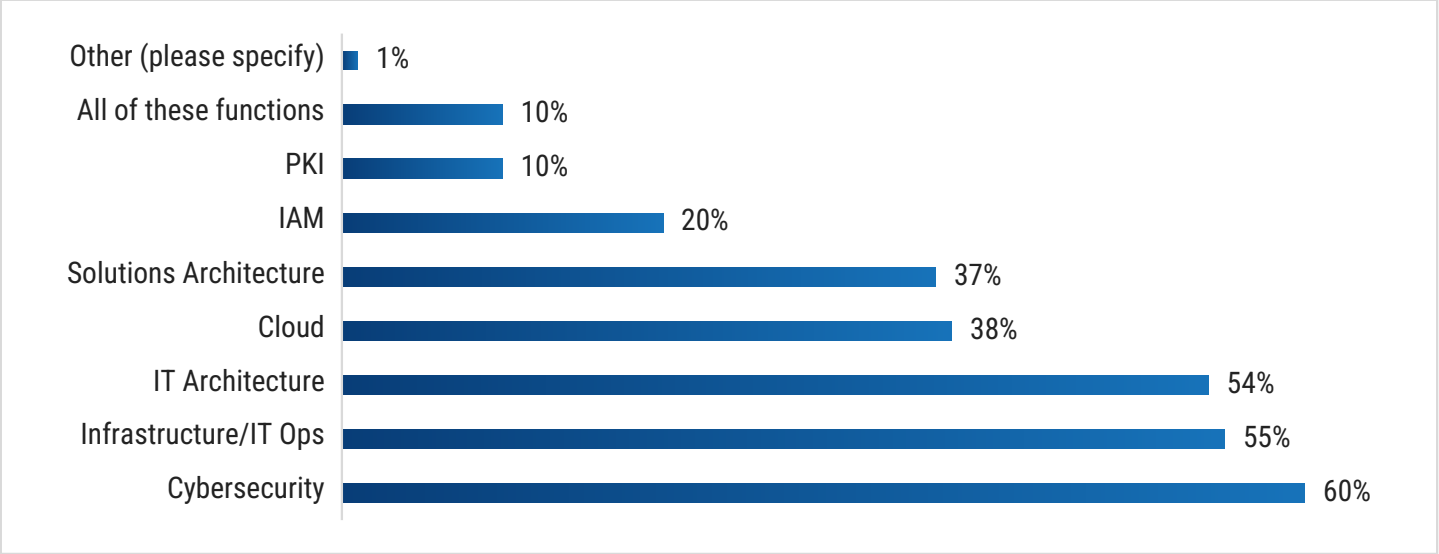
Industry



Organizational level

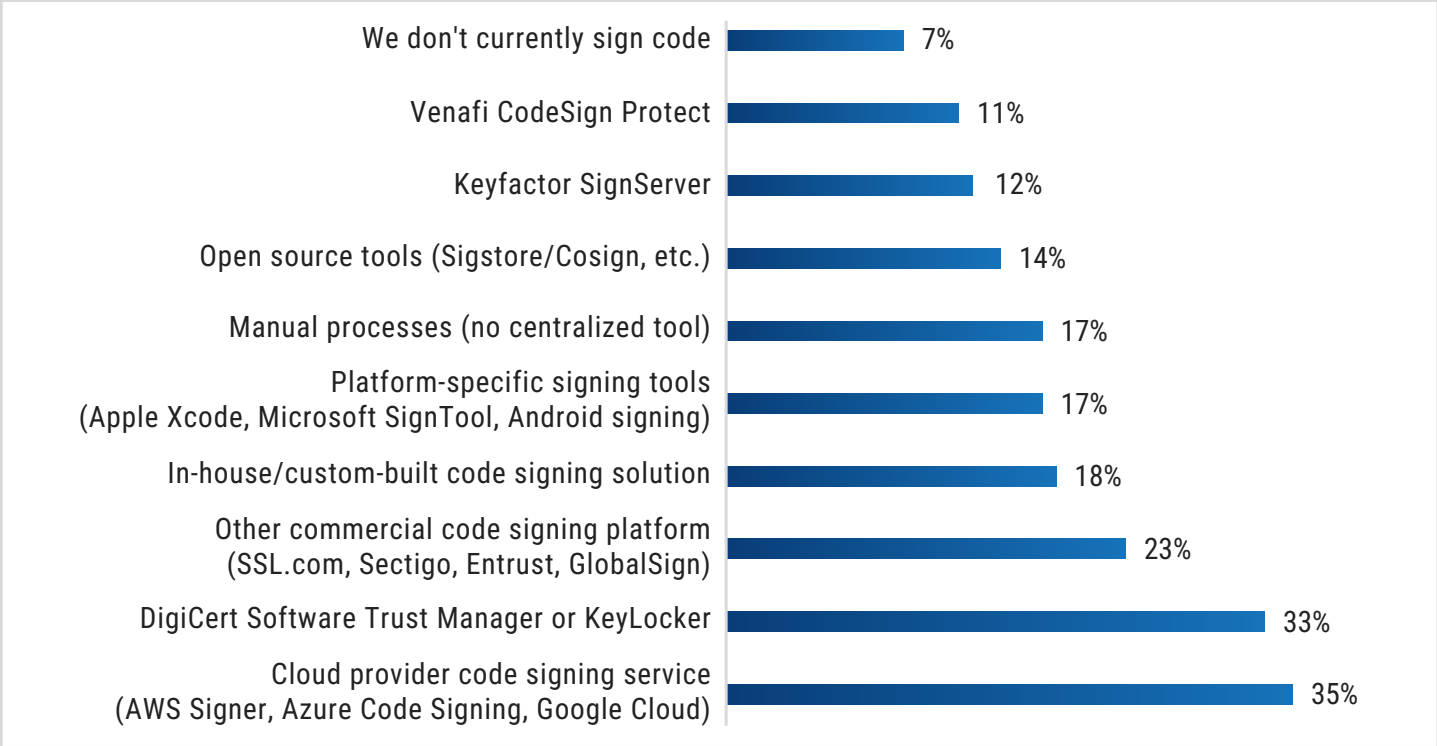


Job function



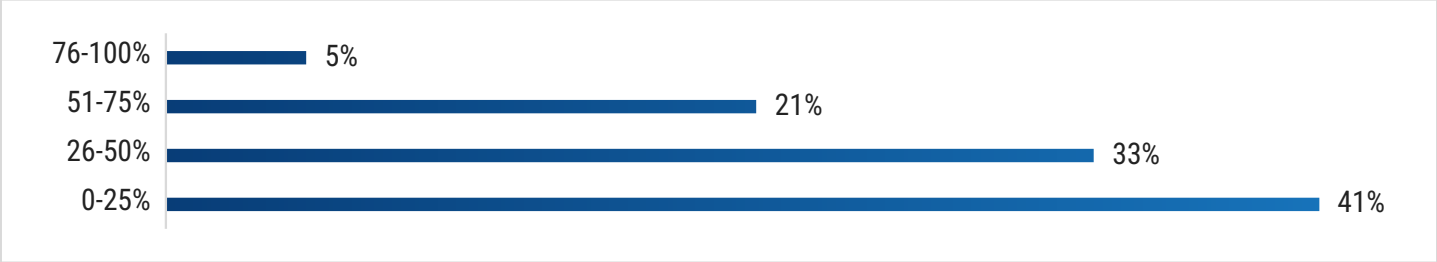
Respondent Profiles: Tools, Practices, and Platforms

Tools used for code signing



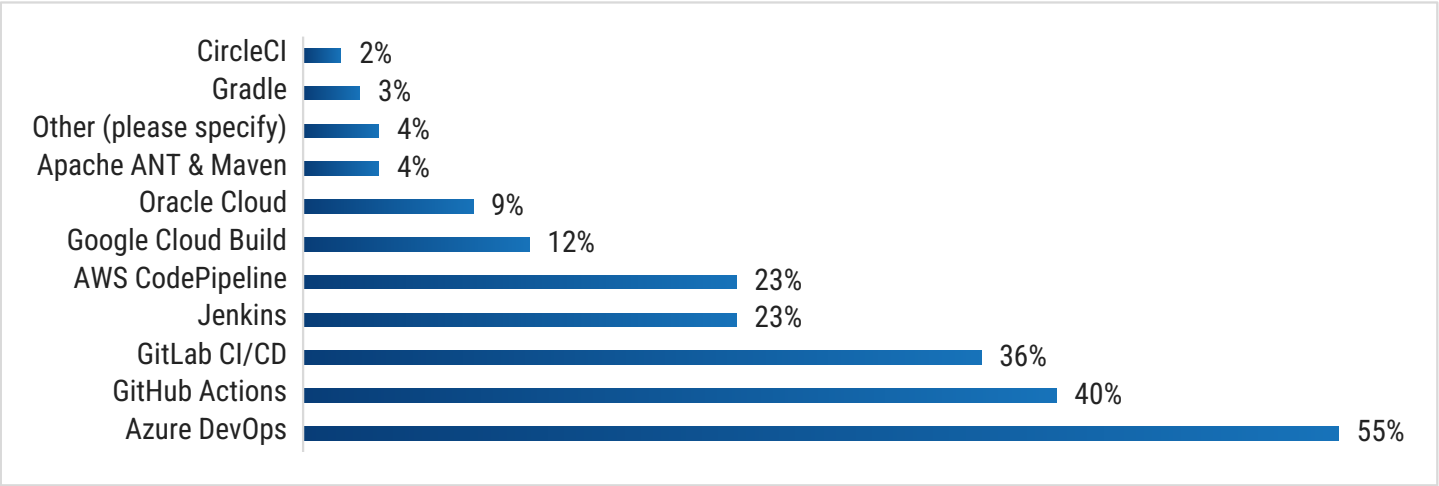
Responses to the question, "Which tools do you currently use for code signing?"

Container/Kubernetes deployment rates



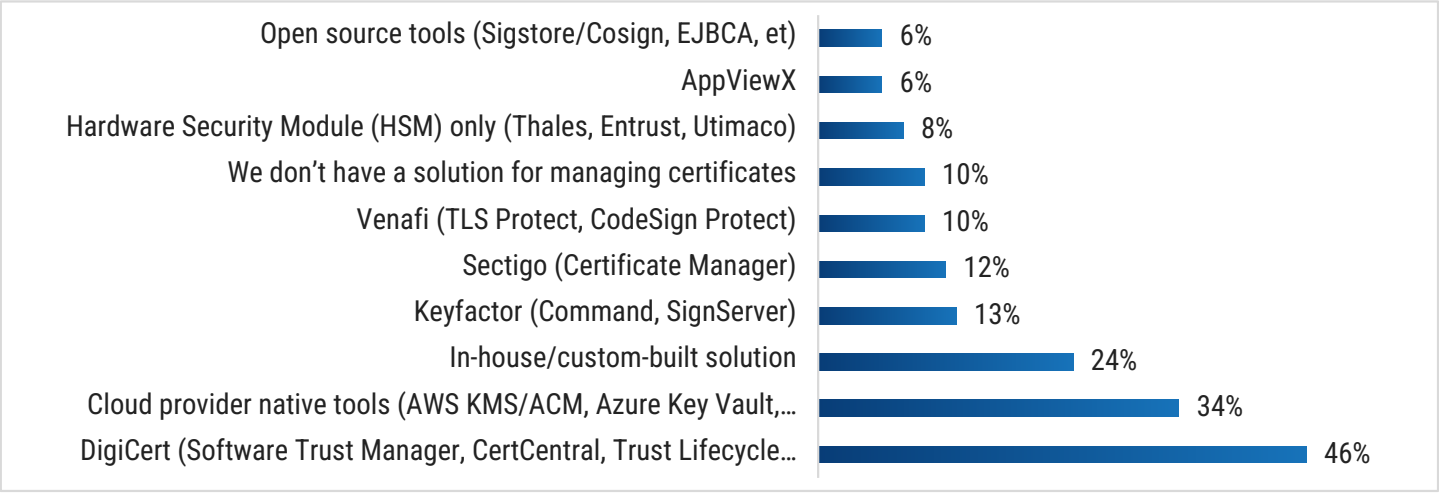
Responses to the question, "What percentage of your applications are deployed in containers/Kubernetes?"

CI/CD platforms used



Responses to the question, "Which CI/CD platforms are you currently using?"

Tools used for code signing



Responses to the question, "Which tools do you currently use for code signing certificate management?"