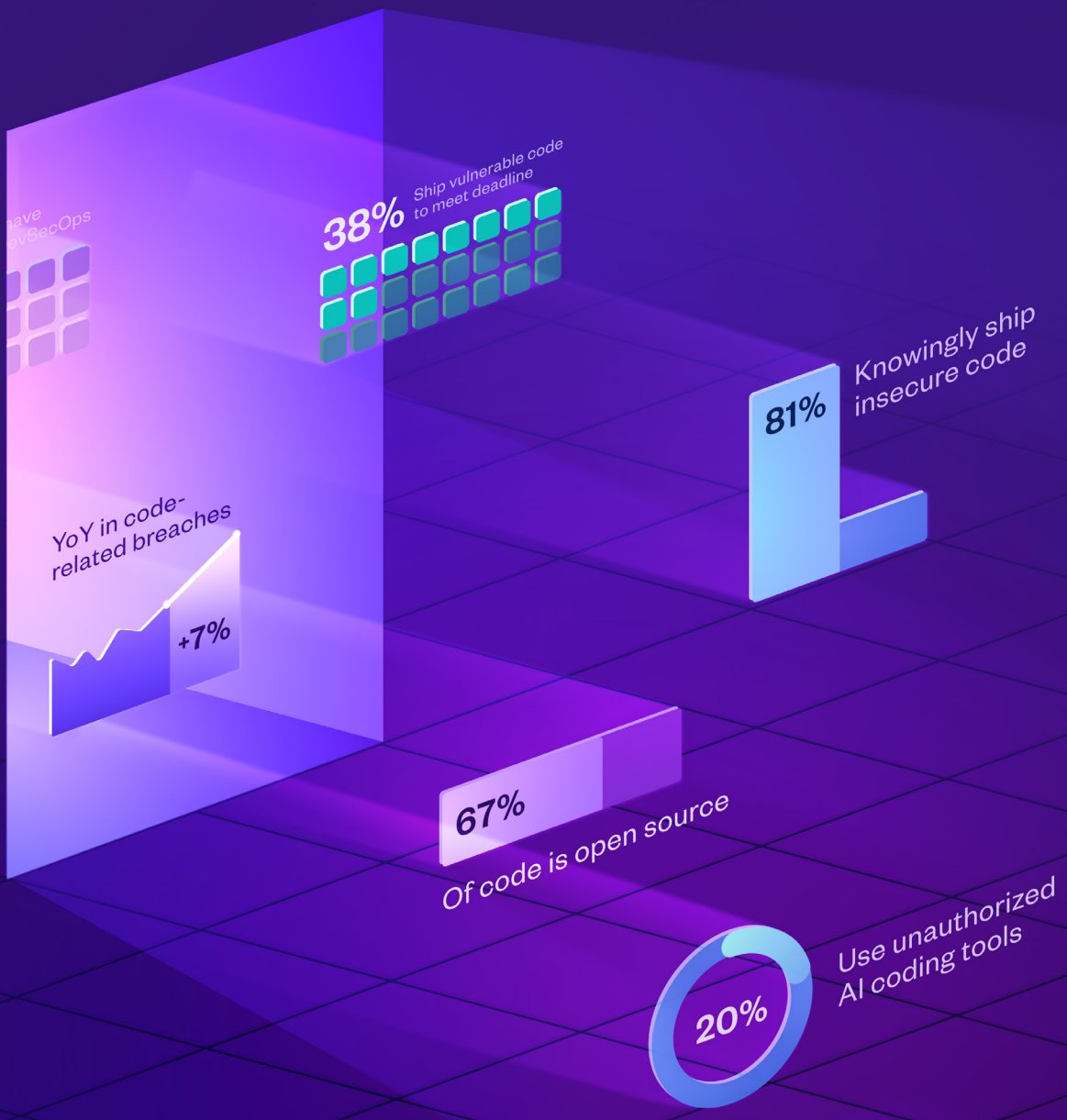
Checkmarx

Report

# The Future of Application Security in the Era of AI

## 2026 Outlook: When a False Sense of Security Meets Breakneck Developer Velocity



# Executive Summary

**2025 marks a turning point in application security. The threat landscape isn't just evolving. It's accelerating faster than most security programs can keep up with.**

Emerging technologies, including AI, multi-cloud architectures, and sprawling software supply chains, are vastly expanding the attack surface.

At the same time, business pressure is forcing teams to knowingly ship vulnerable code, eroding the last guardrails of secure development.

In this environment of increasing pressure to deploy faster, developers turn to AI to write large amounts, if not most of, their code. AI is becoming much more than a tool – it's taking the lead from developers and introducing a new, uncharted frontier of risk.

Application security can no longer afford to be reactive or siloed. It demands immediate, transformative change.

Drawing on insights from over 1,500 AppSec stakeholders globally (CISOs, AppSec Managers, and Heads of Development), this report captures a pivotal moment in the widening disconnect between AppSec awareness and real-world readiness.

It also provides sharp insights on how organizations can not only mitigate but reverse this inertia toward true security maturity that will facilitate and adapt to the upcoming changes.

# Industry Insights

## AI is writing the code. Developers are just hitting deploy, ushering in a new era of risk

As AI-generated code becomes the norm, developer ownership is fading and AppSec is struggling to keep up the pace: While 50% of respondents (excluding Heads of Development) already use AI security code assistants with a third (34%) admitting that over 60% of their code is AI-generated, only 18% have approved usage policies, exposing teams to unmanaged risks and shadow development.

A third of developers admit that
**Over 60% of their code is AI-generated**

## Vulnerabilities are pushed as standard

81% of organizations surveyed admit to knowingly shipping vulnerable code either sometimes or often. This isn't oversight—it's strategy. Under pressure to deliver, teams are treating patch-later practices as acceptable risk, embedding insecurity into the SDLC.

**81% of organizations**
admit to knowingly shipping vulnerable code either sometimes or often.

## Breaches are accelerating—and becoming normalized

A second consecutive year of rise in breach volume indicates deep structural flaws: 98% of organizations experienced a breach from vulnerable code—up from 91% in 2024 and 78% in 2023. The share reporting four or more breaches surged from 16% to 27% year over year, signaling a compounding risk cycle and deeper systemic issues.

**98% of organizations**
experienced a breach from vulnerable code

## Even the most foundational security tools are overwhelmingly underused

Fewer than half of respondents (excluding Heads of Development) are actively using core, mature AppSec tools like DAST (47%) or IaC scanning (48%), despite growing availability.

**50% of respondents**
are actively using core, mature AppSec tools

## DevSecOps talk is common. Execution? Not so much.

Developer responsibility is rising, with vulnerability remediation improving year over year. Still, friction between development and security persists, and many organizations remain in the early stages of adopting DevSecOps holistically. Now is the time to accelerate this cultural shift.

**51% of organizations**
in North America have adopted DevSecOps

# Strategic Imperatives

To close the readiness gap and build a sustainable AppSec posture for 2025 and beyond, **organizations must act decisively:**

01

### Move from awareness to action

Organizations must acknowledge that awareness alone won't stop breaches. A mature security posture requires shared accountability, early intervention, and operationalized practices.

02

### Embed 'code-to-cloud' security

To defend against modern threats, organizations must embed security across the entire software lifecycle. A code-to-cloud strategy ensures continuous protection from initial code creation through CI/CD pipelines to live cloud environments. This full-stack visibility and control is essential for securing complex, distributed applications at scale.

03

### Govern AI use in development

Develop formal policies, approved tools, and auditing practices to secure AI-generated code. Treat AI not just as a risk, but also as a powerful enabler for security automation and intelligent remediation.

04

### Don't just acquire tools, operationalize them

Integrate tools like SAST, DAST, SCA, and ASPM into developer-native workflows and central governance models. Tooling must be unified, measurable, and aligned with business velocity.

05

### Prepare for agentic AI in AppSec

Organizations must begin planning for the integration of **agentic AI** solutions to manage the scale and velocity of AI-generated code. As code volume continues to surge, traditional review and remediation processes won't be enough. AppSec strategies must evolve to include AI-driven agents capable of automating code analysis, policy enforcement, and real-time risk mitigation. In the near future, **the only viable response to AI-scale development may be AI-powered defense.**

06

### Fuel cultural change with developer empowerment

Invest in secure code training, clear security ownership, and incentivized metrics. A culture where developers, security, and operations share responsibility will be foundational to building technical resilience.

# Introduction

**The threat landscape is evolving at an unprecedented pace.**

The convergence of emerging technologies such as artificial intelligence (AI), edge computing, cloud-native development, and complex software supply chains is dramatically expanding the attack surface.

Meanwhile, malicious actors are exploiting this growth with increasing sophistication—automating exploits, scaling social engineering, and leveraging AI to uncover vulnerabilities faster than ever before.

AppSec has always been in a race to catch up with development, managing to stay on pace, but only just. In 2025, however, development is accelerating so rapidly that it's opening a gap too wide for security to bridge without a fundamental shift in mindset.

**We surveyed CISOs, AppSec Managers and Heads of Development to explore their views on:**

- The AppSec readiness gap
- Security risk tolerance
- Security tooling
- AI as a security risk and opportunity
- Security-developer relations

Based on the survey results, this report captures a pivotal moment in the journey of application security.

While awareness of threats is high, the research reveals a maturity gap that leaves many organizations exposed due to cultural inertia, fragmented practices, and outdated security models.

Vulnerable code is knowingly shipped into production at an increasing rate, security tooling remains immature, and emerging risk catalysts—like AI-generated code—are often left ungoverned.

While AI accelerates development, it also introduces new risks, including unvetted code, shadow tooling, and a rise in "vibe coding," where developers trust AI outputs without fully understanding them.

It's time for application security to undergo a major paradigm shift. It is no longer sufficient to bolt on security at the final stages of development or rely solely on isolated tooling. Even shifting left isn't enough – you need to shift security everywhere – including further left, even before your first code commit.

Instead, organizations must adopt modern DevSecOps governance models that make security everybody's responsibility and build the technical resilience needed to embed security across the entire software development lifecycle (SDLC).

Inseparably tied to this shift, organizations should also harness AI not just for code generation, but to accelerate remediation, strengthen Software Supply Chain Security (SSCS), support secure-by-default practices, and improve decision-making throughout the SDLC.

## The Road Ahead: Cautious Optimism

Amid these challenges, signs of progress are emerging. Developers are becoming more engaged in remediation, investment in secure-by-default practices is increasing, and a shift toward code-to-cloud security models is underway.

↘

**The findings in this report** not only highlight where organizations are falling short, but also point to the strategies and structural changes that teams are using to close the AppSec readiness gap, making this more than a snapshot of today's challenges, but a blueprint for securing software in 2025 and beyond.

# Methodology

**Wave 1 2023:**

The research was conducted by Censuswide, among a sample of 517 Software Developers, 534 AppSec Managers and 516 CISOs (aged 18+) across the UK, USA, DACH, ANZ, France, Singapore and Brazil. The data was collected between September 5, 2022 – September 26, 2022.

**Wave 2 2024:**

The research was conducted by Censuswide, among a sample of 1504 Developers, CISOs, and AppSec Managers (aged 18+) across North America, Europe and APAC. The data was collected between November 30, 2023 – December 21, 2023.

**Wave 3 2025:**

The research was conducted by Censuswide, among a sample of 514 CISOs, 501 AppSec Managers, and 504 software developers (aged 18+) across the US, Australia, New Zealand, Singapore, the UK, Austria, Germany, France, and Switzerland. The data was collected between April 7, 2025 – April 29, 2025.

Censuswide abides by and employs members of the Market Research Society and follows the MRS code of conduct and ESOMAR principles. Censuswide is also a member of the British Polling Council.

Chapter 1

# The AppSec Readiness Gap

Despite growing investment and awareness, security breaches caused by vulnerable code remain widespread.

Most organizations know the risks — yet too few are acting decisively to reduce them. The data shows a widening maturity gap: known vulnerabilities are being released, modern threats are poorly defended, and AppSec tooling remains underutilized.

## The cost of complacency

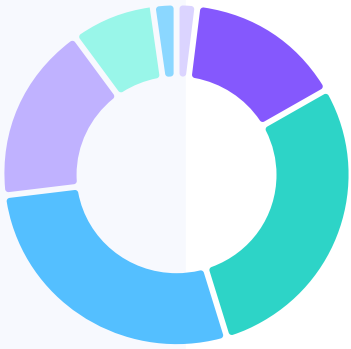
Organizations are facing an expanding and evolving threat landscape and paying a heavy price.

The research reveals that instances of security breaches are on the rise.

98% of respondents say that in the past 12 months their organization experienced one or more security breaches as a direct result of a vulnerable application that their organization developed[1], compared with 91% of those who said the same last year, and 88% in 2023.

The percentage of organizations reporting four or more breaches jumped from 16% in 2024 to 27% in 2025: **an 11-point year-over-year increase.** This drift suggests a compounding risk effect, where each breach potentially weakens defenses, exposes additional vulnerabilities, or signals deeper systemic issues in the organization's software development and security practices.

**Q. In the past 12 months, how many times, if any, has your organization experienced a security breach as a direct result of a vulnerable application that your organization developed?**



| | |
|---|---|
| None | 2% |
| Once | 15% |
| Twice | 29% |
| 3 times | 28% |
| 4 times | 17% |
| 5 times | 8% |
| 5+ times | 2% |

Organizations in Europe (36%) and APAC (34%) are especially likely to have suffered multiple breaches in the past year, with over a third of respondents in each region stating that they have experienced three breaches, while less than a quarter (23%) of those in North America say the same. A 7% YoY increase in breach frequency reveals that despite greater awareness, vulnerability management is not improving. This suggests that policy is disconnected from practice.

The most immediate and visible impact is often business downtime, which 38% of survey respondents cite as an outcome of a breach they experienced in the last 12 months. Typically involving taking infrastructure offline for forensic investigation and remediation, business downtime can severely disrupt productivity and result in a loss of revenue.

(37%) of respondents in the Insurance sector say they have experienced a breach three times in the last 12 months, while just 3% of those in Education say the same.

In parallel, the financial toll (35%) can be substantial, with costs stemming from incident response efforts, regulatory fines, customer compensation, service outages, and increased insurance premiums.

Perhaps even more enduring is the reputational damage (32%) a breach inflicts, as it signifies a failure to fulfill a duty of care, undermines stakeholder trust, and erodes competitive standing. These impacts do not exist in isolation. They compound one another, creating a feedback loop of operational and strategic vulnerabilities. Organizations lacking cybersecurity maturity are especially prone to experiencing the most severe consequences, highlighting the urgent need for proactive and comprehensive risk management.

The findings also indicate that organizations in some industries are more vulnerable than others. Almost 2 in 5 (37%) respondents in the Insurance sector say they have experienced a breach three times in the last 12 months, while just 3% of those in Education say the same.

## ★ What this means

**The False Economy of "Survivable" Breaches**

Organizations have possibly developed a dangerous comfort with security breaches.

But the data shows this mindset is backfiring—98% of organizations experienced breaches in the past year, up 7% from last year. Each breach doesn't just cost money; it can erode customer trust, regulatory standing, and market position in ways that compound over time.

## Actionable Insights by Role:

### CISOs:

Your board has heard breach statistics before and tuned them out. Instead, calculate the cumulative operational drag of your last three breaches—lost deals, extended sales cycles, regulatory audit overhead, customer retention costs. Present security investment as revenue protection, not cost center spending. Get specific about how breaches slow business growth.

### Development Leaders

Audit whether your "developer-friendly" security actually works or just provides cover for shipping vulnerabilities. Create hard stops for critical vulnerabilities in CI/CD and give developers fast remediation paths instead of easy workarounds.

### AppSec Managers:

Opt for measuring developer engagement with security, rather than tool deployment. Low adoption means poor workflow integration. Embed security feedback directly into IDEs and pull requests where developers already work. Track time-to-fix vulnerabilities, not just vulnerability discovery rates.

### The Core Problem:

Organizations knowingly ship vulnerable code, treating this as acceptable business practice. This isn't about needing better tools or more training—it's about normalizing dangerous shortcuts. The gap between security awareness and security maturity is widening because awareness without operational accountability is meaningless.

## Modern threats, same old AppSec

Market trends are putting organizations at risk of diverse security threats.

Respondents cite a wide range of security threats that are expected to cause disruption in the next 12-18 months. These are directly driven by some of the most dominant market trends in software development, which are introducing new vulnerabilities to the SDLC.

Respondents are most likely anticipating software supply chain compromises (35%) and third-party vendor/partner security incidents (35%). This is unsurprising given that 67%[2] of respondents say half or more of their organization's application code consists of open-source software, reflecting the prevalence of modular development and heavy reliance on open-source components, which carry with them their own vulnerabilities.

**67%**

say half or more of their organization's application code consists of open-source software

The market trend toward cloud-native and multi-cloud environments has also increased the likelihood of cloud infrastructure misconfiguration (34%), which ranks among the most anticipated types of security breaches.

Meanwhile, the rise of remote/hybrid working has resulted in distributed workforces and privileged access sprawl, driving concerns around insider threat and privileged access misuse (33%).

Finally, the prevalence of API-first development and digital transformation is increasing the likelihood of API security breaches and business logic attacks, which almost a third (32%) of respondents think their organization is most likely to experience in the next 12-18 months.

### Top five most anticipated types of security breach for the next 12-18 months

1. Software supply chain compromise (35%)
2. Third-party vendor/partner security incident (35%)
3. Cloud infrastructure misconfiguration (34%)
4. Insider threat or privileged access misuse (33%)
5. API security breach or business logic attack (32%)

These findings reflect real trends: as development becomes more modular and distributed, the weakest link — whether a misconfigured Kubernetes pod or a compromised package — becomes the entry point. Yet many organizations still focus AppSec only on internal code, leaving the rest of their SDLC exposed.

### ★ What this means

Organizations are facing a fundamental mismatch: their security threats have evolved to encompass the entire software ecosystem, but their AppSec programs rely on disconnected point solutions that can't correlate risk across the modern development landscape and provide a unified view of risk, with ASPM both in the security dashboard and in the IDEs, so developers know what to fix first. This creates dangerous blind spots where threats slip through the gaps between tools.

**The platform security imperative:**

Traditional point solutions operating in isolation can't address interconnected threats. A compromised open-source dependency (affecting 67%[3] of codebases) might pass code scanning but create supply chain risk, while cloud misconfigurations can expose secure applications, and API vulnerabilities can undermine business logic regardless of code quality. Organizations need integrated platforms that correlate findings across all security layers—like Application Security Posture Management (ASPM)—providing unified risk assessment from initial development through production deployment.

The key is correlation—understanding how a code vulnerability, combined with a misconfigured cloud service and an exposed API endpoint, creates compounded risk that individual tools miss.

## Actionable Insights by Role:

### CISOs:

Stop treating AppSec as disconnected tool management. Invest in platforms that correlate findings across code, infrastructure, APIs, and supply chain to provide unified risk visibility rather than managing security tool sprawl.

### Development Leaders:

Your teams are assembling applications from multiple sources—internal code, open-source libraries, cloud services, and third-party APIs. Security tooling must provide integrated feedback that matches this reality, not separate reports from isolated tools that developers can't prioritize effectively.

### AppSec Managers:

Map your current tool coverage against the top five threat vectors and assess whether tools communicate findings. If you're getting separate alerts from SAST, cloud scanning, and dependency monitoring without understanding combined risk, you're creating alert fatigue instead of actionable intelligence.

### The Core Insight:

Modern applications are assembled, not just written. Security programs using disconnected point solutions are securing individual components while missing the systemic risks that emerge when those components interact.

## Modern threats, same old AppSec

Organizations are not prepared to face emerging threats but continue to put themselves at risk.

Despite a clear awareness of the diverse and growing threat landscape, many organizations lack the maturity, visibility, and controls needed to effectively mitigate these risks across today's complex software development ecosystems.

Under pressure to move faster and deliver more, teams are increasingly cutting corners, often sidelining security in favor of speed. This trade-off leaves critical gaps in protection, exposing organizations to threats they know exist but feel ill-equipped to fully address.

One such trade-off is the fact organizations without plans to adopt software supply chain security (SSCS) are more likely to face repeat (two or more) security breaches from vulnerable in-house apps than those using this software (89% vs 78%).

SSCS is a sign of maturity too, as those using it are more likely to have highly mature application security posture (34%).

Organizations are especially unlikely to be prepared to defend against the following potentially disruptive security challenges:

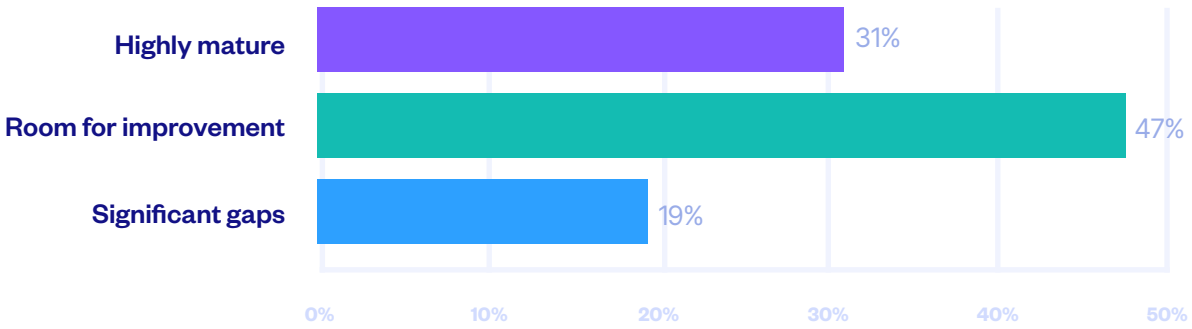
1. Attacks targeting CI/CD pipelines or development environments (14%)
2. Securing emerging technologies (IoT, edge computing etc.) (14%)
3. Supply chain compromises and upstream dependency attacks (14%)
4. Advanced API security threats and business logic exploits (13%)
5. Managing security implications of generative AI in development workflows (12%)

Less than 15% of organizations are prepared for threats that have become mainstream in the last two years. These include CI/CD attacks, business logic exploits, and unmanaged AI risk — all of which can bypass traditional AppSec controls.

The research highlights specific weaknesses in organizations' security posture, which further findings indicate are symptomatic of an overall lack of maturity.

Just 31% of CISOs and AppSec managers say their security posture is highly mature and industry-leading. However, 47% state that although they consider theirs above average, there is room for improvement, while, concerningly, almost a fifth (19%) of respondents admit that there are significant gaps in their security posture.

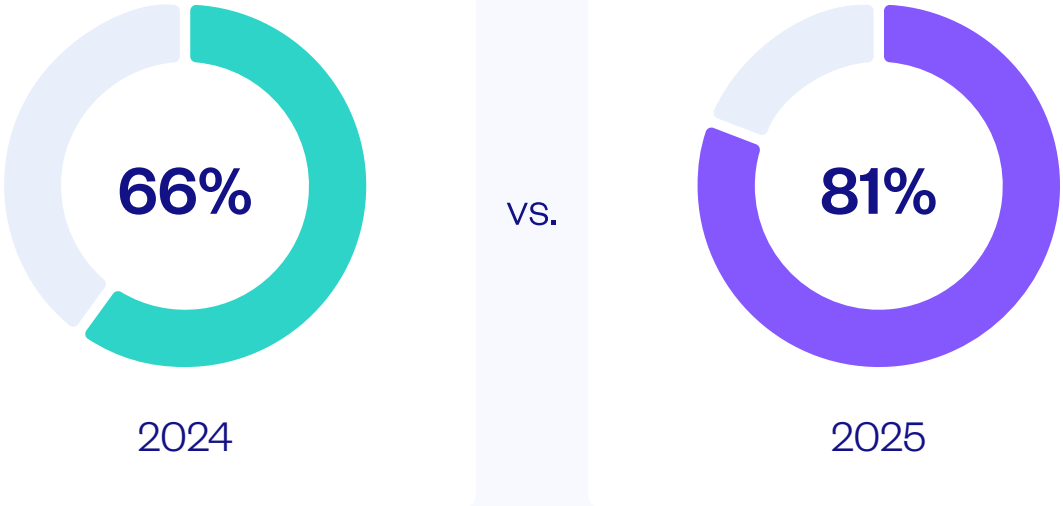
## Application security posture ratings




These security gaps may exist because development and security teams are under relentless pressure to meet business goals. However, even with a clear understanding of where their defenses fall short, many organizations are still knowingly pushing exploitable code into production, choosing speed over security, despite the risks.

In 2024, 15% of respondents said their organization (or an organization they know) often deployed known vulnerable code into a production environment, which rose to 26% of respondents in this year's survey. Meanwhile, the percentage of respondents who say they believe this sometimes happens has risen from 51% to 55%.

## % of organizations who deployed known vulnerable code into a production environment often or sometimes[1] in...



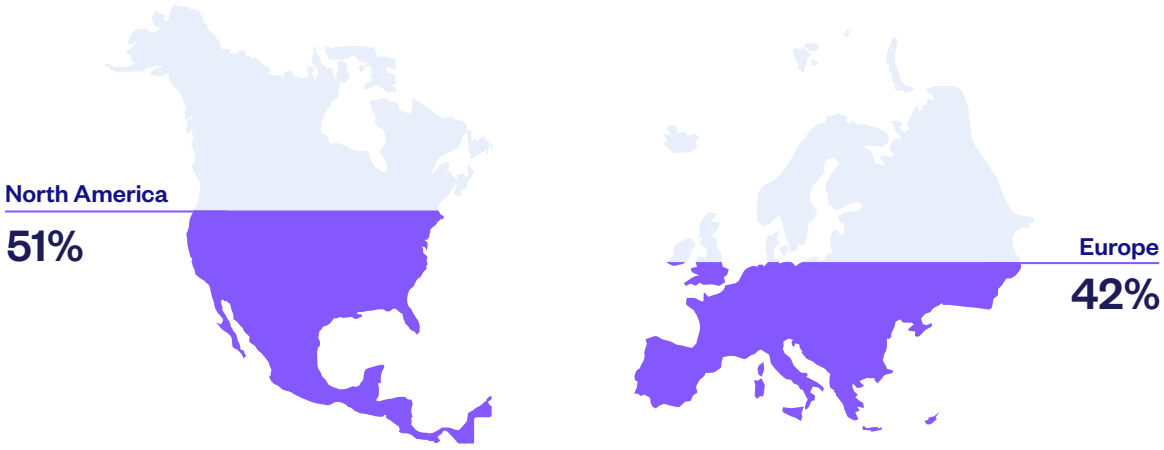
This data, coupled with the fact that known and/or unknown vulnerability in code released to production (35%) is cited as the top reason for security breaches that have occurred in the past 12 months, reflects a growing resignation among developers that truly risk-free code is unrealistic in the context of today's fast-paced business environment.

The findings indicate that organizations in Europe are most likely to be engaging in this risky behavior. Almost a third (32%) of respondents in this region say their organization (or an organization they know) often deploys code with known vulnerabilities, while just 24% of those in North America say the same.

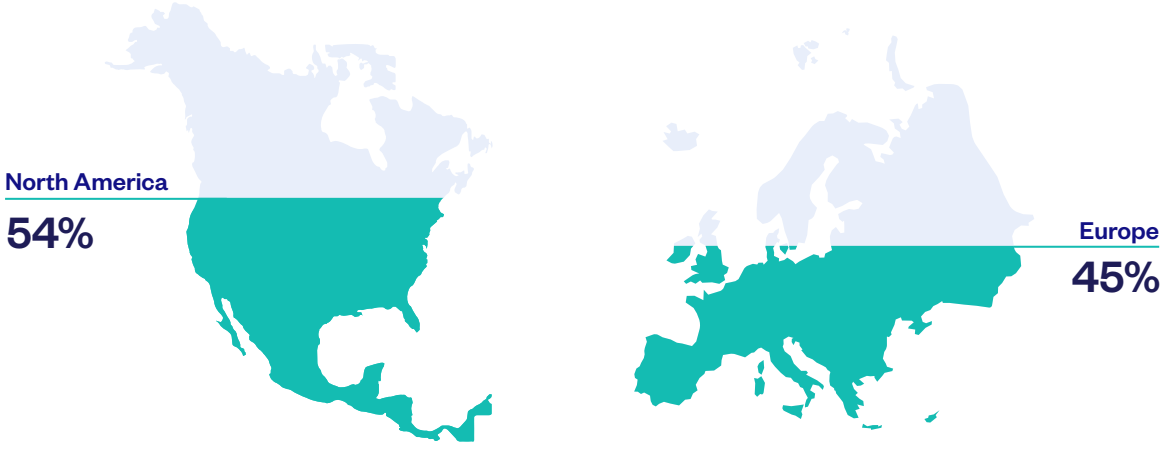
One possible explanation for this difference is that organizations in North America appear to be further along in shifting their security mindset. For instance, 51% of respondents in this region have adopted a DevSecOps approach, compared to just 42% in Europe. Similarly, 54% of North American organizations have implemented secure code training, while only 45% of their European counterparts report doing so, indicating a more proactive investment in building security into development from the ground up.

## % of organizations implementing...

### DevSecOps




### Secure code training



## ★ What this means

Organizations have entered a dangerous phase of informed negligence—they understand the risks but are systematically choosing to accept them. The 15-point jump from 66% to 81% of organizations knowingly shipping vulnerable code reveals that security isn't losing ground to ignorance, but to calculated business pressure. This represents a fundamental breakdown in risk governance.

**The Moving Target Problem:**

The threat landscape is in constant flux—AI's explosion over the past year created entirely new attack vectors that most organizations still aren't prepared for. What seems like adequate security today becomes obsolete as new threats emerge. The 77% of security leaders who rate their posture as mature or above-average may be measuring against yesterday's threats while today's attacks bypass their controls entirely.

## Actionable Insights by Role:

### CISOs:

Implement periodic maturity assessments using structured frameworks like Checkmarx's Application Security Posture Management Assessment (APMA) to benchmark against current threat realities, not past performance.

### Development Leaders:

Implement hard stops for critical vulnerabilities in CI/CD pipelines while providing developers fast remediation paths.

### The Cultural Difference:

The regional differences (32% of European vs. 24% of North American organizations often ship vulnerable code) correlate with DevSecOps adoption rates (42% vs. 51%). This isn't coincidence—organizations that embed security into development culture rather than treating it as a separate function see measurably better outcomes and adapt faster to new threats.

Chapter 2

Systemic Risk Tolerance:

# Why Known Vulnerabilities Reach Production?

The data makes it clear: shipping vulnerable code is not a rare exception — it's a systemic issue. Nearly a third of respondents say they knowingly push insecure code to production, often due to deadline pressure or misplaced confidence in post-release patching. The result? Technical debt disguised as velocity.

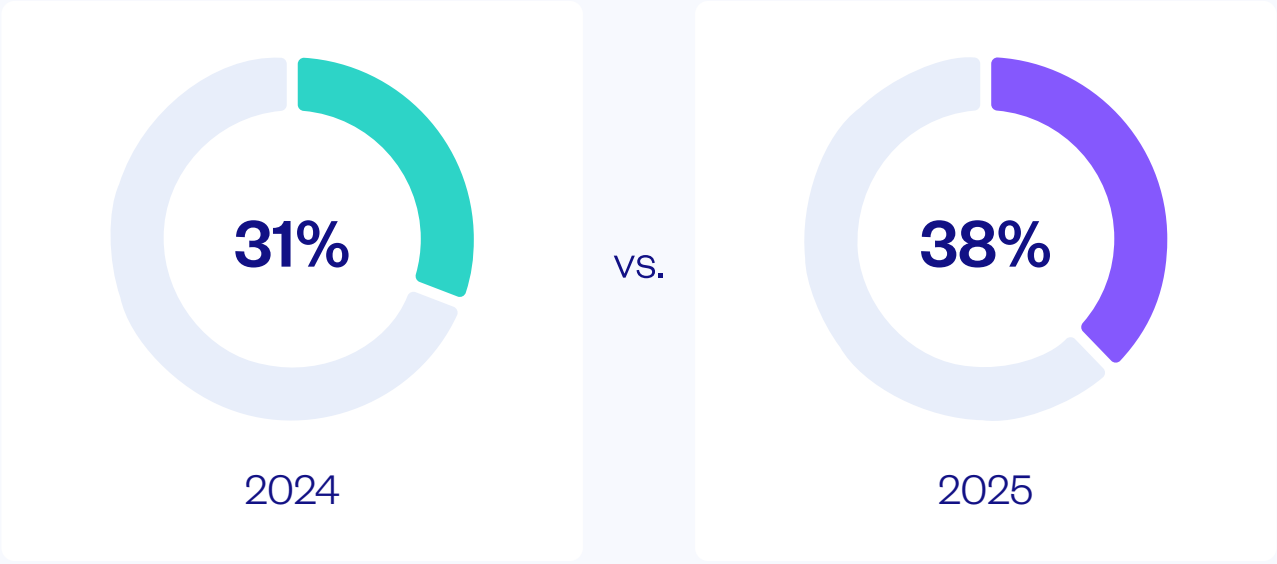
Without earlier intervention, cultural shifts and better tooling, this risky behavior will persist as a major threat to application security. Each vulnerable release creates immediate attack surface while accumulating technical debt that becomes harder to remediate over time.

## Has speed over safety become the new normal?

Organizations are accepting increasing security risks as a trade-off for progress.

In 2025, 38% of respondents state that vulnerable code is being shipped to meet a business, feature or security-related deadline, compared to 31% the previous year.

**% who say vulnerable code was deployed into production to meet a business, feature, or security-related deadline**



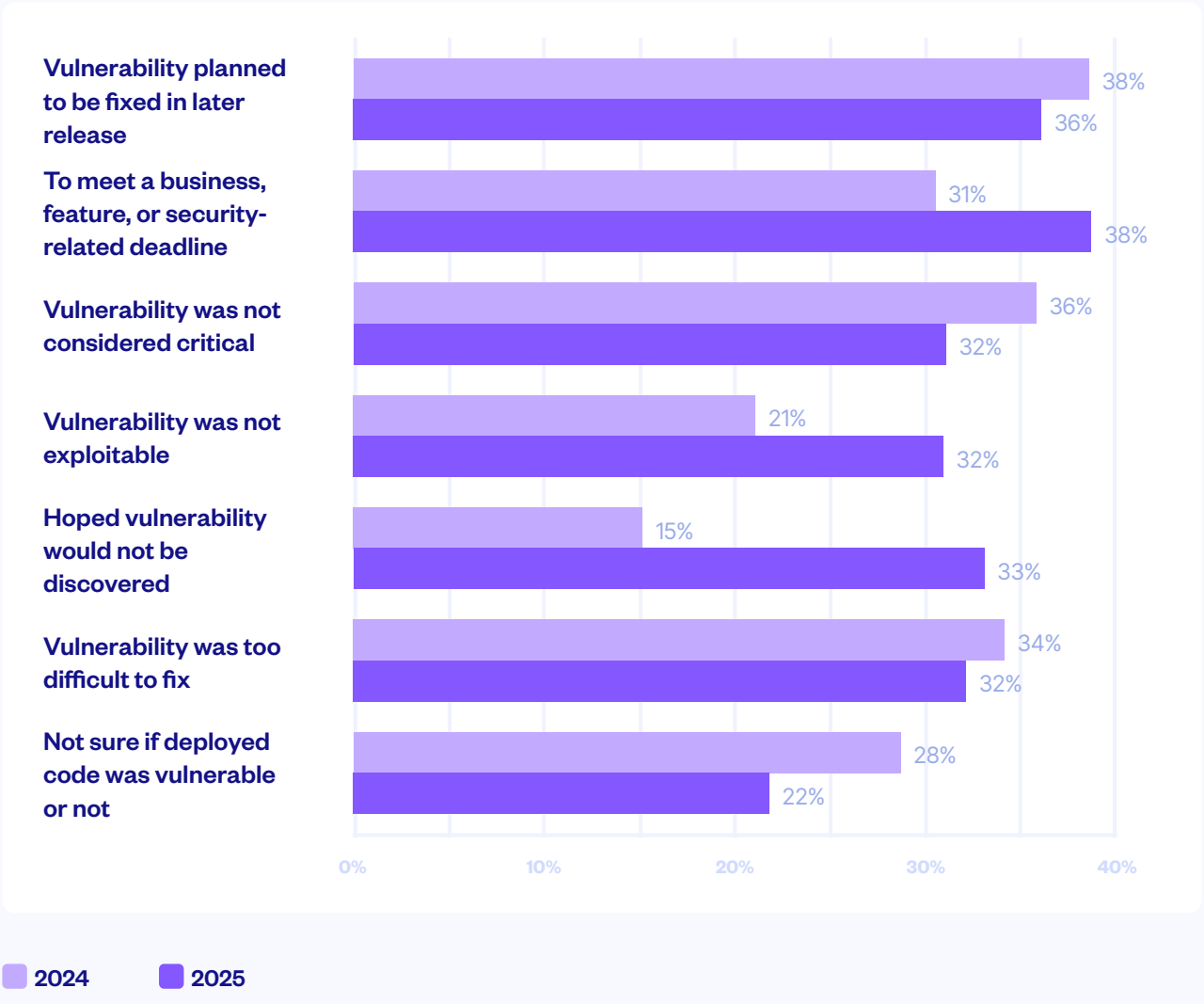
Not only does this highlight that businesses are prioritizing innovation and speed to market over secure code, it also hints at a possible reason for the gaps found in organizations' security postures. It may be that while under extreme pressure to deliver new features and services, security and development teams find themselves lacking the time and resource to address weaknesses in their threat landscape. At the same time, it's possible that C-suites are unaware of the risks that their demanding, business-driven deadlines are introducing. When asked to name reasons for deploying vulnerable code, many respondents also stated that they planned to fix the vulnerability in later release (36%) suggesting that remediation is performed too late in the SDLC. Perhaps most concerning of all, is the fact that a third (33%) of respondents admit that they hoped a vulnerability would not be discovered, showing that organizations are not adequately prioritizing security while under pressure to meet demanding business requirements.

The findings also show that this behavior is becoming more prevalent among developers. In 2024, just 15% said they hoped vulnerabilities in the code they were shipping would not be discovered, which rose to 31% in this year's survey. This striking difference may indicate that development teams are feeling increasingly overwhelmed and may lack the time, tools and resources to fix known issues before deploying vulnerable code.

Shipping now and fixing later has become an implicit policy. But with rising attack automation, even short-lived vulnerabilities can be exploited within minutes of release.

This is institutionalized risk. When almost a third of developers are "hoping vulnerabilities won't be discovered," it suggests a cultural problem — not just a tooling gap. Security must be incentivized as part of delivery success, not treated as a blocker.

## Q. Why was vulnerable code deployed into production?




This shift reflects cultural strain: without shared responsibility, developers are left to choose between shipping and securing.

★ What this means

This is not a tooling issue alone — it's a governance failure. If insecure releases are the norm, then AppSec needs a cultural reset, not just more scans. Organizations should consider adopting Google DORA metrics. Organizations must embed security metrics into delivery goals and reward secure code, not just speed.

This data reveals a critical governance breakdown: shipping vulnerable code has become institutionalized policy disguised as pragmatic decision-making. The 18-point spike in developers "hoping vulnerabilities won't be discovered" (from 15% to 33% between 2024 and 2025) isn't just about time pressure—it's evidence of a cultural collapse where security responsibility has been abandoned rather than shared.

## Actionable Insights by Role:

### CISOs:

This is less of a tooling challenge and more of a governance challenge requiring cultural intervention. Implement security metrics as delivery success criteria alongside traditional performance indicators. Adopt frameworks like Google's DORA metrics that measure both delivery velocity and operational stability, including security outcomes.

### Development Leaders:

The 38% of teams shipping vulnerable code to meet deadlines are creating unsustainable technical debt. Implement "definition of done" criteria that include security validation. Fix the workflow to make secure delivery the easier path, not the exception.

### AppSec Managers:

The 33% of developers hoping vulnerabilities won't be discovered represents a complete breakdown in security culture. This requires immediate intervention—not more training or tools, but process changes that make security validation automatic and fast. Create shared visibility dashboards that show both development and security leaders the same vulnerability data in real-time.

### The Governance Reset Required:

Organizations must fundamentally restructure incentives. Security must be embedded into delivery success metrics, with rewards for secure code delivery rather than just speed. This requires executive alignment on redefining what "successful delivery" means.

**The Bottom Line:**

The normalization of shipping vulnerable code under deadline pressure represents systematic risk tolerance that compounds over time. Each vulnerable release doesn't just create immediate security debt—it reinforces a culture where security corners can be cut when business pressure mounts. This approach is unsustainable in an environment where attackers are increasingly automated and persistent.

Chapter 3

# From Critical Exposure to Technical Resilience

Tools are improving — but they're still not embedded deeply enough. Only around half of respondents use mature technologies like DAST, container security, or IaC scanning. Leaders are investing — but even those with mature postures aren't fully protected.

## Tools in hand, but not in play

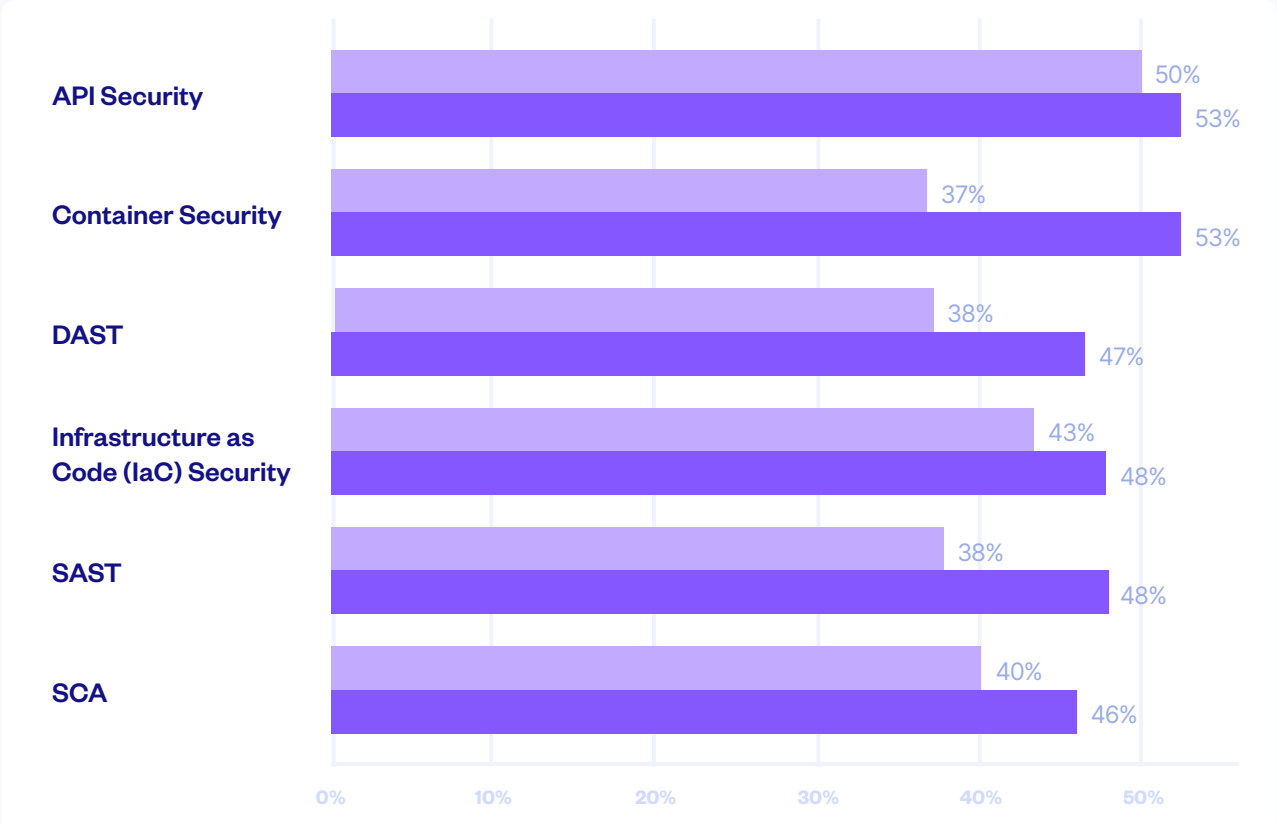
Organizations are on their way to achieving technical resilience but are not yet taking advantage of all the security tools available to them.

Despite knowingly releasing vulnerable code, many organizations may be leaving parts of their SDLC exposed.

In fact, relatively low percentages of organizations are using security approaches that have been around for a long time, including API security (53%), container security (53%) and DAST (47%).

A comparison with last year's findings shows that while there have been some marginal improvements in tooling adoption, gaps remain.

### Q. Which of the following application security technologies/approaches do you have?




% of CISOs and AppSec Managers who said they use this AppSec technolog

2024 2025

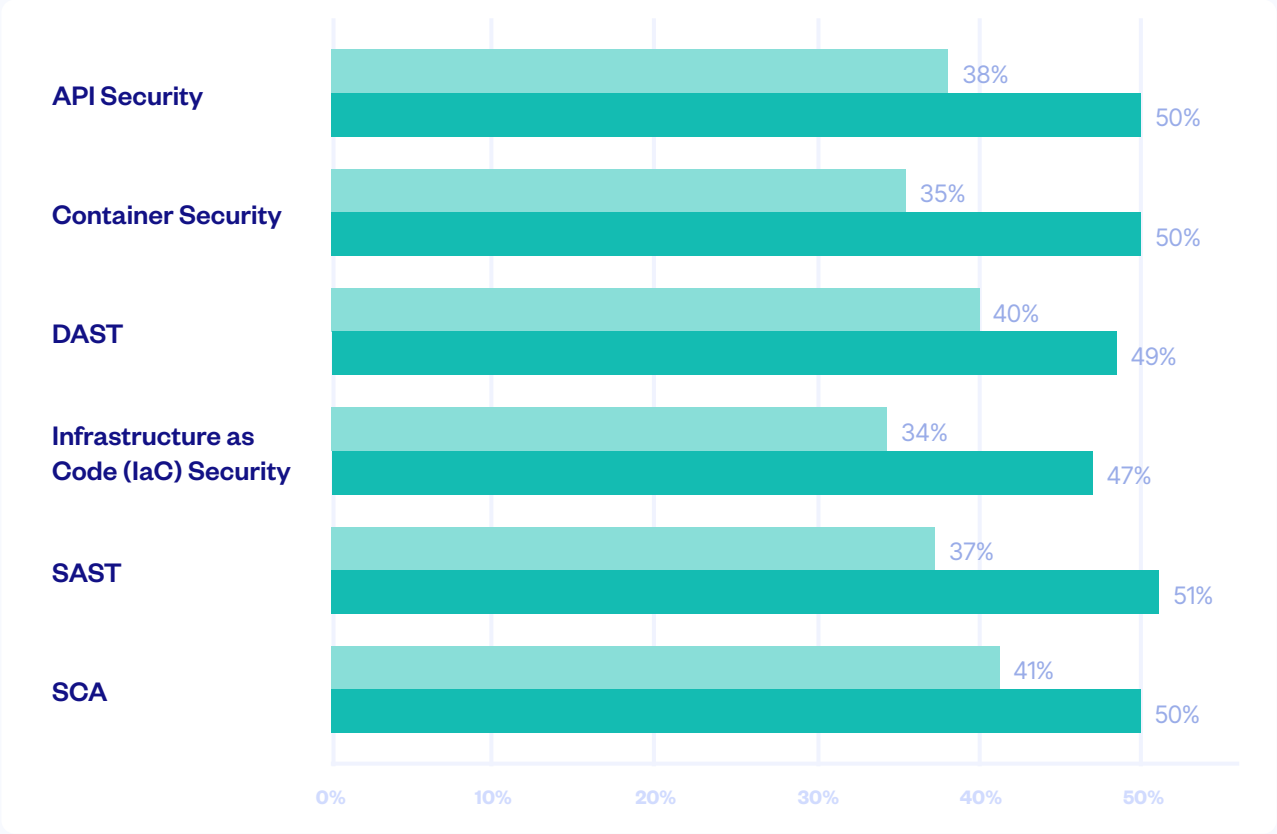
Despite the growing availability of security tooling — from SAST and DAST to software supply chain controls — most organizations are still underutilizing even the basics. According to Gartner's Hype Cycle for Application Security, most core AppSec tools have passed the "Peak of Inflated Expectations" and are well into the "Slope of Enlightenment" or beyond. In other words, the technology is no longer the issue — adoption is.

Yet our survey shows only around half of organizations are actively using DAST (47%), container security (53%), or Infrastructure-as-Code scanning (48%). API security testing, software bill of materials (SBOM), and policy-as-code remain similarly under-deployed. Even for tools considered mainstream, real-world implementation is behind.

This lag reveals a critical insight: the barrier isn't access — it's integration. As AppSec tools mature, the complexity of managing them across fragmented CI/CD environments has become a limiting factor. To fully realize their value, these tools must be operationalized — unified into developer-native workflows, governed through central platforms like ASPM, and measured through actionable KPIs.

Encouragingly, the findings show that more organizations are planning to invest in more tools in future. What's more, a comparison with data from our previous survey suggests that they are likely to follow through with this as the percentages of CISOs who say their organization uses various tools has grown year on year.

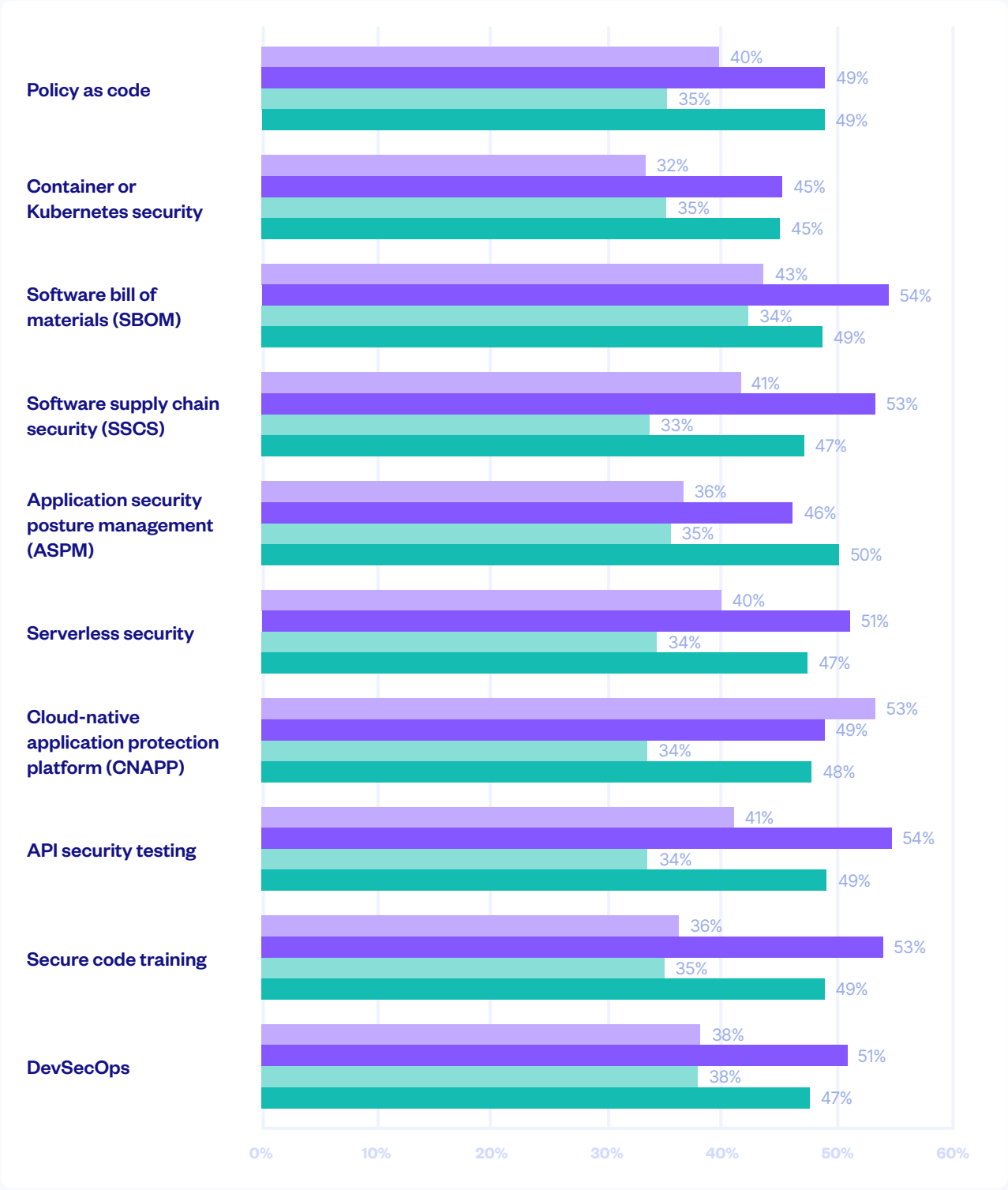
## Q. Which of the following application security technologies/approaches do you have?




Organizations are also showing a willingness to invest in additional tooling to help bolster their security posture. Significantly, this includes new and emerging security technology such as AI security code assistants, which 50% of respondents say they are already using, while a further 40% say they plan to invest in this in the next 2-3 years. Meanwhile, 48% of respondents say they are already using Application Security Posture Management (ASPM) while a further 40% say they plan to use this technology in the next 2-3 years.

Once again, the research suggests that this investment is happening as adoption rates for these technologies has grown year on year.

## Q. Which of the following technologies do you use?




The findings reveal the tools that organizations with the most advanced and well-established security practices are more likely to have embedded in their security posture.

These include:

- Container security (56%)
- DAST (49%)
- IaC security (51%)
- Malicious Package Protection (49%)
- SAST (54%)
- SCA (51%)

Meanwhile, these organizations are also more likely to be using the following technologies and approaches:

- Policy as code (51%)
- AI security code assistants (54%)
- Penetration testing as a service (53%)
- Container or Kubernetes security (50%)
- Software bill of materials (SBOM) (54%)
- SSCS (55%)
- Serverless security (53%)
- Secure code training (52%)

One area that it is vital for all organizations to prioritize is DevSecOps. Currently, just under half (49%) of respondents say they have already adopted this approach. A further 39% say that they are planning to in the next 2-3 years. But, given how critical DevSecOps is to achieving a mature and resilient security posture that's capable of defending against modern cyber threats, any organization that has yet to do so should be focusing on embedding this philosophy across every stage of their SDLC now.

## ★ What this means

The security tool landscape has reached a paradox: technologies are mature and available, but adoption lags behind threat evolution. Despite proven effectiveness, fundamental tools remain underutilized, creating a false sense of security where organizations believe they're protected simply because tools exist in their environment.

**The Integration Crisis:**

The barrier has shifted from "what tools should we buy?" to "how do we make them work together?" Organizations are accumulating security technologies without operational coherence, creating alert fatigue instead of actionable intelligence.

## Actionable Insights by Role:

### CISOs:

The fact that 48% are already using ASPM platforms shows organizations are moving beyond tool sprawl toward unified security visibility.

### AppSec Managers:

The adoption benchmarks from mature organizations provide a clear implementation roadmap: container security (56%), SAST (54%), SBOM (54%). These aren't aspirational targets—they're operational requirements for defending against current threats.

### Development Leaders:

Developer adoption of security tools correlates directly with workflow integration quality. The 90% of organizations investing in AI security assistants (50% current, 40% planned) indicate demand for security that enhances rather than interrupts development velocity.

### The DevSecOps Imperative:

Only 49% have implemented DevSecOps, with 39% planning adoption over 2-3 years. This timeline mismatches threat velocity—modern attacks exploit vulnerabilities within hours of deployment.

**The Reality Check:**

Organizations can no longer treat security tool adoption as a multi-year journey. The correlation between security maturity and comprehensive tool utilization demonstrates that technical resilience requires immediate, orchestrated implementation of available technologies, not gradual expansion of disconnected capabilities.

Chapter 4

# AI: The AppSec Curse and (Possibly) the Cure

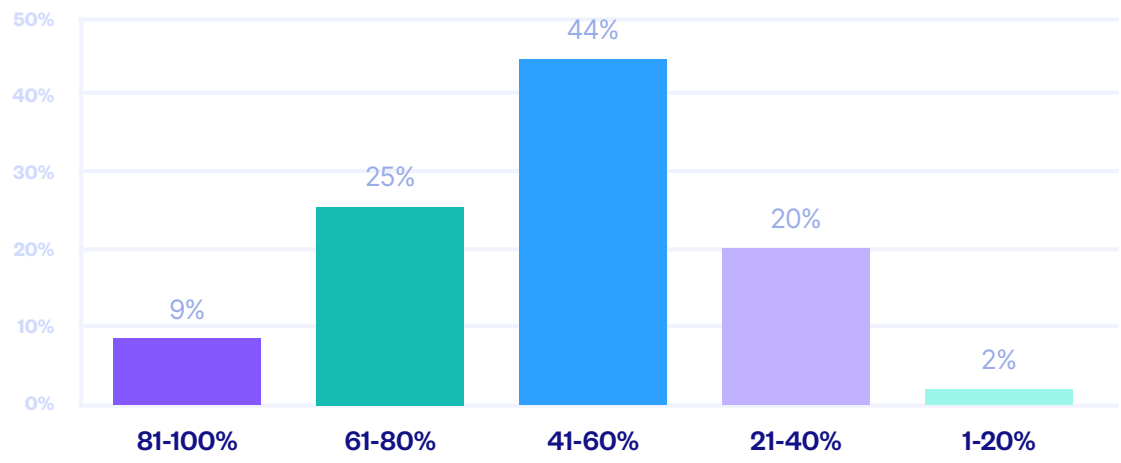
AI is no longer a future-facing concept in application security. It is a defining force shaping the present and future of development and defense.

As organizations grapple with an increasingly dynamic and complex threat landscape, AI presents both a powerful opportunity and a significant risk vector. Its integration into development and security workflows has the potential to enhance technical resilience, streamline remediation, and enable intelligent decision-making, but only if it is strategically governed and embedded with care.

## AI has infiltrated the SDLC – can it help to secure it too?

The research reveals that organizations are already heavily reliant on AI-generated code. 44% of respondents report that 41–60% of their organization's code was AI-generated in 2024. Meanwhile, 25% say this figure is even higher, between 61–80%.

### Q. What percentage of your organization's code do you think was AI-generated in 2024?

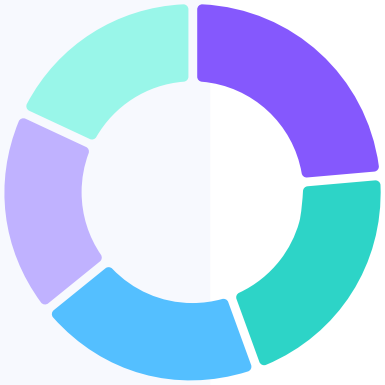



However, while AI enables scale and speed, it also introduces new attack surfaces and governance gaps, particularly when usage is not formally approved or monitored. This appears to be the case in a worryingly large proportion of organizations.

A fifth (20%) of respondents say AI code generation tools are not permitted yet know or assume that these tools are being used without authorization. This rises to 26% in larger organizations with 5,000–10,000 developers, indicating widespread Shadow AI and Shadow Development.

Meanwhile, less than a fifth (18%) of respondents say their organization has a list of approved tools, suggesting that the majority of organizations do not have complete oversight of AI usage. This lack of visibility could pose a significant challenge to security teams as vulnerabilities introduced by AI-generated code may go undetected and unremediated; a critical concern for organizations striving toward mature security postures.

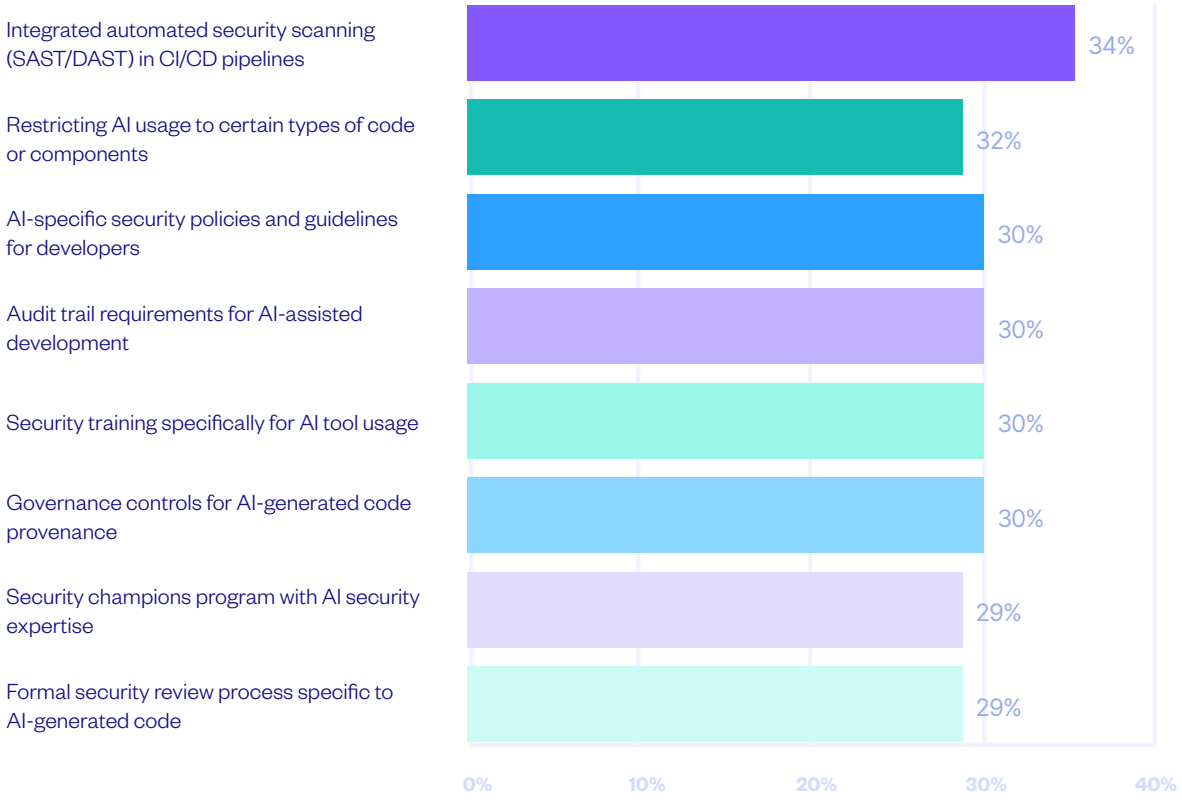
## Q. Which statement best reflects your organization's position on AI tools for code generation?



- **24%** Different groups within the greater organization are allowed to purchase different tools on a case-by-case basis
- **21%** Individuals are allowed to use different tools on a case-by-case basis
- **20%** AI code generation tools are not permitted for use in my organization, but I know or assume that individuals are using them without permission.
- **18%** AI code generation tools are not permitted for use in my organization, and I don't know of anyone using them without permission.
- **18%** We have a list of approved tools

The findings also show that despite the extensive use of AI, very few organizations have specific protocols in place to ensure code written by AI coding assistants such as GitHub Copilot, Qodo Gen and Cursor AI is secure.

## Q. How does your organization ensure code written by AI coding assistants is secure?



The shift is underway: 50% of respondents now use AI code assistants, with another 40% planning to invest. But few have implemented policies, audit trails, or secure usage guidelines. Shadow AI is now as real a threat as Shadow IT was a decade ago.

Security teams are already overwhelmed by the sheer volume of potentially vulnerable code flooding in. As AI continues to scale and accelerate development, that pressure is only set to intensify.

Paradoxically, AI itself may prove essential to overcoming this challenge.

Indeed, AI is viewed as an ally in strengthening security programs. When integrated effectively, AI offers valuable support across a range of security operations.

## Top five things respondents would trust AI used in a security tool to do

01 **Act as an AI assistant to answer developers' questions (40%)**

02 **Provide just-in-time training for developers on identified vulnerabilities (39%)**

03 **Analyze security data to provide recommendations (39%)**

04 **Implement remediation steps after human approval (37%)**

05 **Outline remediation options for humans to implement (33%)**

AI is also seen as a proactive enabler of security excellence. 29% of respondents believe AI has the biggest potential to enhance their security posture by helping to tailor AppSec solutions to the nuances of different applications and 25% cite that it has a role to play in accelerating vulnerability remediation. Meanwhile, 23% highlight AI's potential to correlate data and simplify complex security analysis.

The findings show that organizations are already aware that AI is not just a productivity tool but strategically essential to advancing the maturity and technical resilience of their security posture.

## ★ What this means

AI has created a fundamental asymmetry: organizations are rapidly adopting AI code generation (70% report 41-80% of code is AI-generated) while governance lags behind. Only 18% have approved tool lists, and 20% know AI tools are being used without authorization. This represents a massive expansion of unmanaged risk—essentially conducting development with ungoverned assistants that may introduce vulnerabilities at scale.

**The Shadow AI Crisis:**

The 26% of large organizations (5,001-10,000 developers) experiencing unauthorized AI tool usage signals a governance breakdown similar to Shadow IT a decade ago, but with accelerated risk velocity. When developers use unapproved AI tools, security teams lose visibility into code provenance, making vulnerability tracking and remediation exponentially harder.

## Actionable Insights by Role:

### CISOs:

AI governance can't wait for perfect policies. Implement immediate controls: real time in-IDE SAST scanning, approved tool lists, audit trails for AI-assisted development, and mandatory security scanning for AI-generated code. The 50% using AI assistants with 40% planning adoption means this window for proactive governance is closing rapidly.

### Development Leaders:

Shadow AI usage indicates your developers need AI tools but lack approved options. Provide sanctioned AI assistants with built-in security guardrails rather than forcing underground usage. Implement code provenance tracking to identify AI-generated sections for targeted security review. Make sure you are running SAST scans at the beginning, including real-time AI Secure Coding Assistants.

### AppSec Managers:

AI presents both a challenge and a solution. While AI-generated code expands your attack surface, AI security assistants can help manage the scale. The 39% who trust AI to provide just-in-time vulnerability training and 39% who see value in AI-driven security recommendations show appetite for AI-powered security operations. Start with AI assistants for developer education and remediation guidance.

### The Strategic Opportunity:

Organizations viewing AI only as a risk are missing its defensive potential. AI can correlate security data (23% see this potential), accelerate remediation (25%), and tailor AppSec solutions to application-specific contexts (29%). The key is implementing AI security tools at the same velocity as AI development tools.

**The Time Pressure:**

With AI code generation already representing the majority of code in many organizations, security teams face an immediate scaling challenge. Traditional manual security review processes cannot keep pace with AI-accelerated development.

Chapter 5

# The Security-Development Rift Is Closing — Slowly

Security champions may be growing in number, but the old friction between security and development is far from solved. Developers are expected to own security — but without shared context, training, or aligned metrics, the burden often leads to shortcuts.

## Are developers upping their security game?

A lack of alignment between security and development teams is a persistent challenge organizations face when it comes to application security.

**CISOs and AppSec Managers in particular cite various issues relating to this strained relationship among their biggest organizational challenges when it comes to application security, including:**

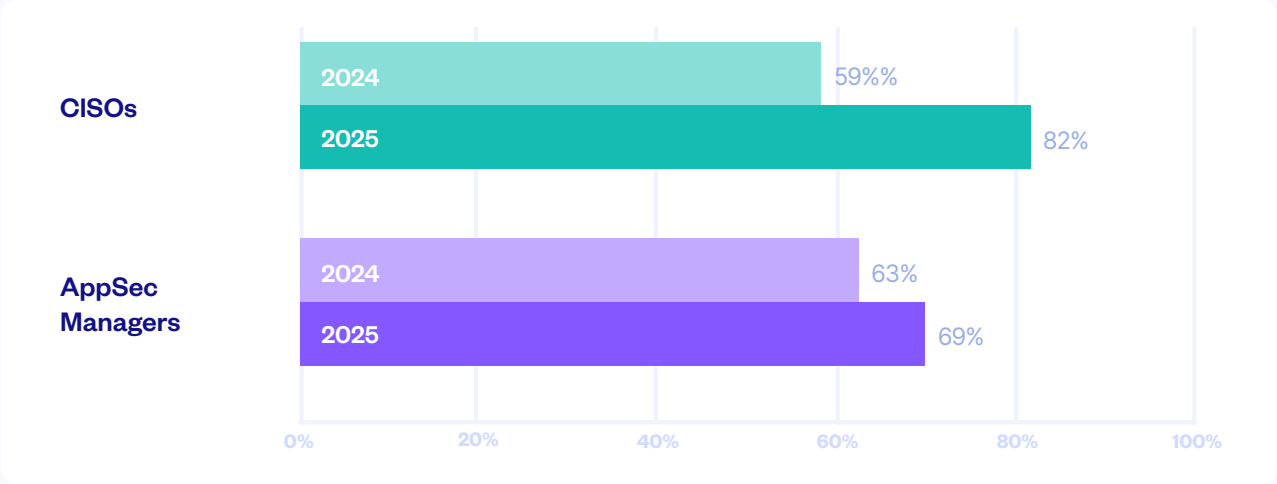
- Scaling security practices across multiple development teams (18%)
- Difficulty integrating security into existing CI/CD pipelines (16%)
- Insufficient security training for development teams (15%)
- Developers don't prioritize security requirements (12%)
- Poor communication between security and development teams (11%)

These frustrations are understandable. After all, developers are knowingly shipping vulnerable code, putting organizations at risk. However, it's important to remember that they are doing so while under extreme pressure to meet business requirements for constant innovation and speed to market.

Fortunately, the data indicates that developers are taking more responsibility for security.

Comparing data from our 2024 survey with this year's results reveals that both CISOs and AppSec Managers recognize developers' increased contribution to security practices.

## % of CISOs and AppSec Managers who said over half of vulnerabilities were fixed by developers in 2024 vs 2025




This increase warrants closer examination – especially with CISOs. One interpretation is that fewer vulnerabilities float "up" to the CISOs, so AppSec see a modest improvement since they have a broader view of vulnerabilities, while CISOs see that overall the 'serious' vulnerabilities. Fixed improved drastically – it indicates a different mindset but also that devs prioritize better maybe.

These numbers suggest positive momentum — but the perception gap between CISOs and AppSec managers is a red flag. Without shared data and aligned tooling, remediation ownership becomes a guessing game.

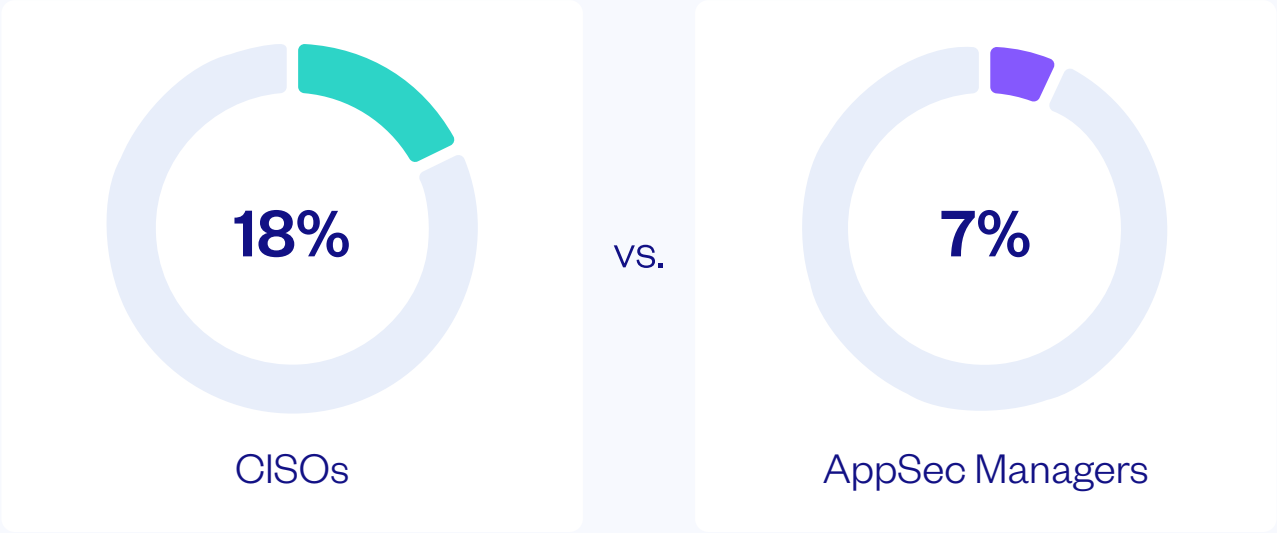
These findings are a promising sign that security is becoming better integrated within the SDLC and that developers are becoming more security aware, which is a positive step towards improving the dynamic between security and development functions.

However, there is also evidence to suggest that the security-development relationship isn't the only one that could be improved.

The findings reveal a disconnect between CISOs and AppSec managers on the percentage of vulnerabilities being fixed by developers.

Currently, 18% of CISOs think 91%-100% of vulnerabilities are fixed by developers, while 7% of AppSec managers say the same. Meanwhile, 23% of CISOs think 76%-90% of vulnerabilities are fixed by developers, compared to 32% of AppSec managers who say the same.

## % who think 91%-100% of vulnerabilities are fixed by developers



## ★ What this means

The dramatic jump in CISO perception of developer remediation (59% to 82%) versus the modest AppSec manager increase (63% to 69%) reveals a critical visibility gap that could undermine security programs. This divergence suggests CISOs are seeing fewer escalated vulnerabilities reach their level, while AppSec managers maintain ground-truth awareness of the full vulnerability landscape.

**The Perception Problem:**

When 18% of CISOs believe developers fix 91-100% of vulnerabilities while only 7% of AppSec managers agree, it indicates either improved developer triage of critical issues or dangerous blind spots in executive visibility. CISOs may be seeing better outcomes for high-severity vulnerabilities while missing the broader pattern of accumulated technical debt from unaddressed lower-priority issues

## Actionable Insights by Role:

### CISOs:

Your optimistic view of developer remediation may reflect improved escalation filtering rather than comprehensive security improvement. Implement shared dashboards with AppSec teams that show the full vulnerability spectrum, not just critical issues reaching executive attention. The gap between your perception (82%) and AppSec reality (69%) suggests you're missing important security debt accumulation.

### AppSec Managers:

The CISO perception gap creates both opportunity and risk. CISOs seeing fewer escalated issues may provide political cover for security investments but could also lead to complacency about ongoing remediation challenges. Create executive reporting that shows both wins (critical issues resolved by developers) and persistent challenges (volume of unaddressed vulnerabilities).

### Development Leaders:

The increase in developer remediation responsibility (evident across both metrics) requires better tooling integration and training support. If AppSec teams still see significant remediation gaps while CISOs perceive improvement, developers need clearer prioritization guidance and faster feedback loops to address the full vulnerability spectrum effectively.

### The Risk:

This perception misalignment could lead to premature declarations of victory while fundamental security-development integration challenges remain unaddressed.

# The Path to Security-Development Alignment

The security-development relationship is evolving, but organizations must move beyond cultural initiatives toward operational transformation. The data shows promise: 44% report increased DevSecOps investment and 39% cite more developer security training. Yet the persistent friction points—scaling security practices (18%), CI/CD integration challenges (16%), and insufficient training (15%)—reveal that investment alone isn't enough.

**The Integration Challenge:**

Modern applications require security coverage across the entire technology stack—from initial code development through cloud deployment. This comprehensive approach addresses the reality that today's threats exploit vulnerabilities anywhere in the software lifecycle: compromised dependencies in the supply chain, misconfigured cloud infrastructure, vulnerable APIs, and insecure code. Organizations need unified platforms that provide visibility and control across to all these layers rather than managing disconnected security tools.

**Developer Empowerment, Not Burden:**

The dramatic increase in developer remediation responsibility must be supported with better tooling and workflows. True developer-friendly security means embedding security feedback directly into existing development environments—IDEs, pull requests, CI/CD pipelines—rather than requiring separate security processes. When security validation is automatic and provides clear remediation guidance, developers can address vulnerabilities without disrupting their delivery velocity.

**The Cultural and Technical Bridge:**

DevSecOps represents more than shared responsibility—it requires shared visibility and aligned incentives. The perception gap between CISOs and AppSec managers on vulnerability remediation highlights the need for unified metrics and dashboards that give all stakeholders the same view of security progress. Organizations must measure secure delivery velocity, not just delivery speed or security coverage independently.

**Optimism Grounded in Action:**

Organizations expecting security improvement over the next 12-18 months reflect realistic confidence based on ongoing investments. However, success depends on treating security as a delivery enabler rather than a gate. Organizations achieving this transformation embed security validation into every development decision point while providing developers with tools, training, and incentives that make secure choices the easier choices.

The future of application security lies not in resolving the tension between speed and security, but in eliminating it through integrated platforms and aligned workflows that make secure development the natural path forward.

# Conclusion

The application security landscape in 2025 represents a fundamental paradigm shift that extends far beyond traditional cybersecurity concerns. What we're witnessing is the emergence of security as a core business capability that directly impacts competitive advantage, operational resilience, and market positioning.

The convergence of AI-accelerated development, cloud-native architectures, and increasingly sophisticated threats has created a new reality: security can no longer be treated as a compliance checkbox or risk mitigation exercise. Organizations that continue to view security through this lens will find themselves at a systemic disadvantage against competitors who have embedded security as a delivery accelerator.

### The Strategic Imperative:

The most successful organizations in the coming years will be those that recognize security as a strategic differentiator. When security is deeply integrated into development workflows, it becomes a source of velocity rather than friction—enabling faster, more confident releases while reducing the operational overhead of incident response, regulatory scrutiny, and technical debt remediation.

### The Cost of Delay:

The gap between security-mature organizations and those still treating security as afterthought will widen exponentially. As AI reshapes software development and attackers leverage automation to scale their operations, the organizations stuck in reactive security models will face escalating costs—not just from breaches, but from slower time-to-market, reduced developer productivity, and lost competitive opportunities.

### The Network Effect of Security Maturity:

Organizations with mature security practices create positive feedback loops that compound over time. Developers become more productive when they receive clear, actionable security guidance. Business leaders gain confidence to pursue ambitious digital initiatives when they trust their security foundation. Customers and partners increasingly view security maturity as a prerequisite for engagement, creating market advantages for security-forward organizations.

**The transformation required isn't just operational—it's philosophical.**

The future belongs to organizations that stop asking "how do we secure what we build?" and start asking "how do we build security into everything we do?" This shift from security as oversight to security as enablement will define the winners and losers in an increasingly digital economy.

# Key Takeaways

## For AppSec Managers

**Operationalize and embed tools into developer workflows**

Despite having access to mature technologies like SAST, DAST, IaC scanning, and SBOMs, most organizations underutilize them. AppSec managers must focus on integrating these tools natively within CI/CD pipelines, aligning with developer processes, and leveraging platforms like ASPM to enable unified visibility and control across the SDLC. Tooling is only as effective as its adoption and integration.

## For CISOs

**Shift from policy to practice and nurture a security-aligned culture**

While 98% of organizations experienced breaches due to vulnerable code, 81% knowingly shipped that code, often to meet business goals. This reveals a systemic disconnect between policy and execution. CISOs must drive a cultural transformation that embeds security into delivery metrics, aligns incentives across functions, and ensures that speed does not come at the cost of risk. Governance must go beyond compliance to become a strategic enabler of resilience.

## For Developers

**Own security as part of quality, not an afterthought**

Developers are fixing more vulnerabilities than ever before yet are still often expected to choose between shipping and securing. With 33% admitting they hoped vulnerabilities wouldn't be discovered post-release, it's clear that security is still seen as a blocker rather than a shared goal. Embracing DevSecOps and code-to-cloud practices means integrating secure coding habits, leveraging AI responsibly, and collaborating closely with security teams to build software that's both fast and safe.

**Notes:**
[1]'Once', 'Twice', '3 times', '4 times', '5 times' and 'More than 5 times, please specify' responses combined. | [2]'Often' and 'Sometimes' responses combined. | [3]'50%-74%', 75%-99% and '100%' responses combined.