2026 STATE OF SOFTWARE SECURITY

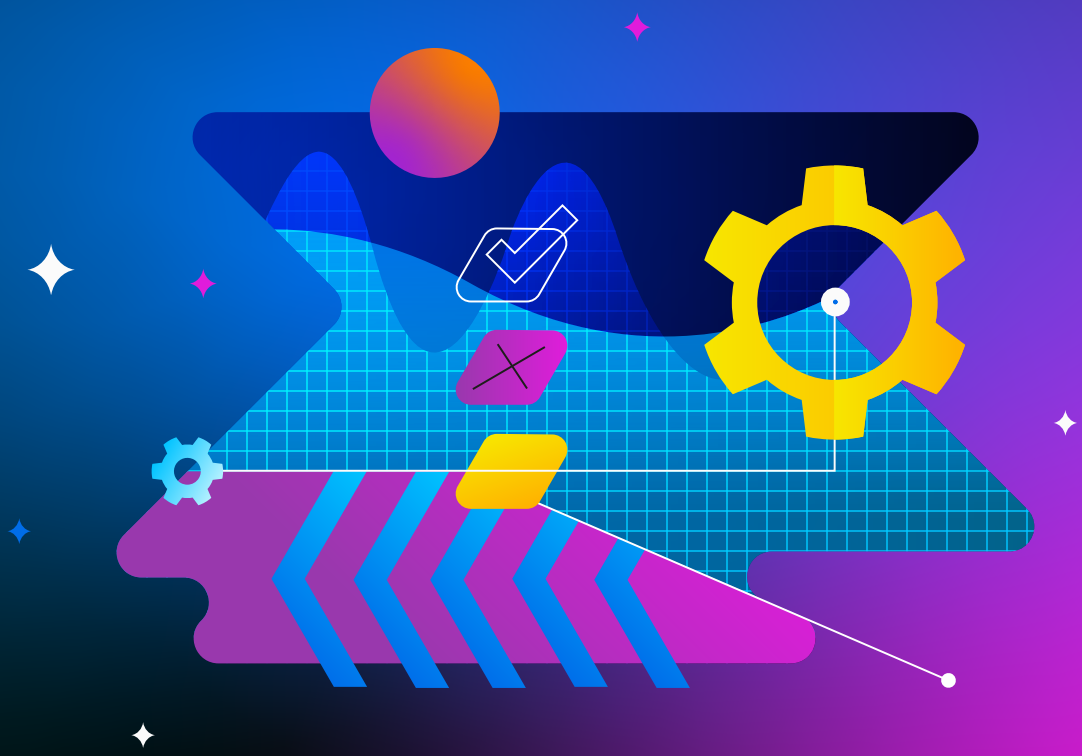
PRIORITIZE, PROTECT, PROVE

VERACODE

Contents

Executive Summary

What happens when keeping pace isn't an option?

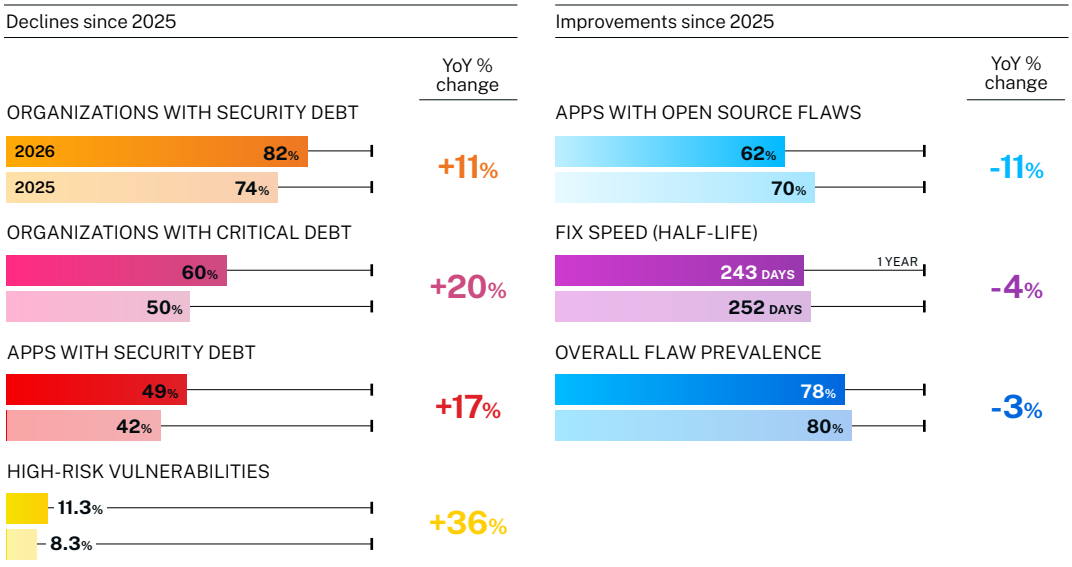


Innovation and risk are inseparable. As organizations build the future, they will inevitably create vulnerabilities. The critical question is not how to eliminate all risk, but which risks are you willing to accept?

The 2026 State of Software Security (SoSS) report illuminates a difficult truth: the pace of flaw creation is decisively outstripping the current capacity for remediation. Despite marginal gains in fix rates, the tide of security debt – known vulnerabilities left unresolved for more than a year – is rising. This is not a distant problem; it is a present reality for 82% of organizations, an 11% increase in a single year. Plus, the debt accumulating is not benign. Critical security debt (flaws that are both severe and highly exploitable) now affects 60% of organizations, a stark 20% rise from the previous year, and high-risk vulnerabilities saw a 36% relative increase.

When the velocity of development in the AI era makes comprehensive security unattainable, the strategy must evolve. The path forward is not about running faster on a treadmill of endless flaws. It's about making deliberate, intelligent choices about which risks to accept and which to neutralize. It's about learning to prioritize, protect, prove, and, ultimately, prevail.

This summary analysis puts 2026 findings against the 2025 baseline to illuminate the primary themes that will shape our understanding of software security maturity in an era where AI-driven development, expanding attack surfaces, and accelerating release cycles collide with finite remediation capacity. The following chart measures not just the absolute change, but the percentage of change year-over-year (YoY). This captures, for example, how a shift from 50% to 60% critical debt reflects a 20% relative increase, not just a 10-point rise. This approach provides a more nuanced view of the rate at which risk indicators are accelerating or improving over time.

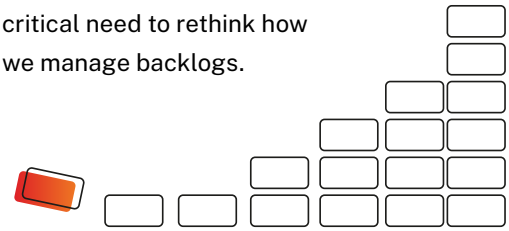


Key themes

As we delved into the findings for this year's report, we saw multiple themes arise:

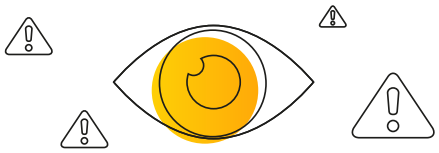
The Security Debt Crisis Intensifies

With 82% of organizations now burdened by security debt, the accumulation of vulnerabilities older than a year is outpacing remediation capacity, signaling a critical need to rethink how we manage backlogs.



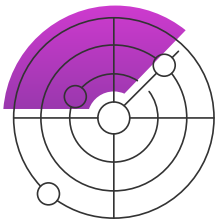
The High-Risk Vulnerability Surge

A 36% YoY spike in flaws that are both highly severe and likely to be exploited demands an urgent shift from generic severity scoring to prioritization based on real-world attack potential.



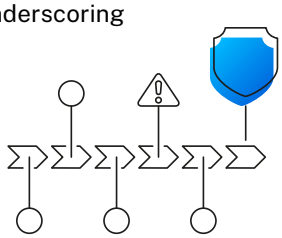
Modest Progress in Detection, Struggles in Remediation

While organizations are successfully finding fewer flaws and improving detection rates, the data reveals a persistent struggle to fix them quickly enough to close the widening exposure window.



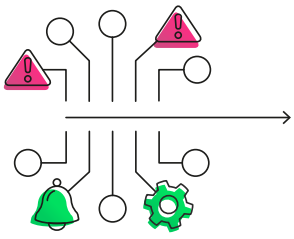
The Persistent Third-Party Supply Chain Challenge

Despite improvements in general open-source hygiene, third-party components remain the primary source of critical, long-lived debt, underscoring the necessity of rigorous supply chain defenses.



The AI Era's Double-Edged Impact

The rise of AI is reshaping the landscape by potentially introducing new patterns of high-risk vulnerabilities while simultaneously offering the automated remediation capabilities needed to finally turn the tide.



CHAPTER 1

The Security Debt Crisis Intensifies

Security debt—known vulnerabilities left unresolved for more than a year—has surged dramatically, with organizational prevalence climbing from 74% to 82% and rising from 50% to 60% of organizations in a single year. This trend has continued year after year, with security debt mounting and affecting more and more organizations.

Key Points

Overall Security Debt:

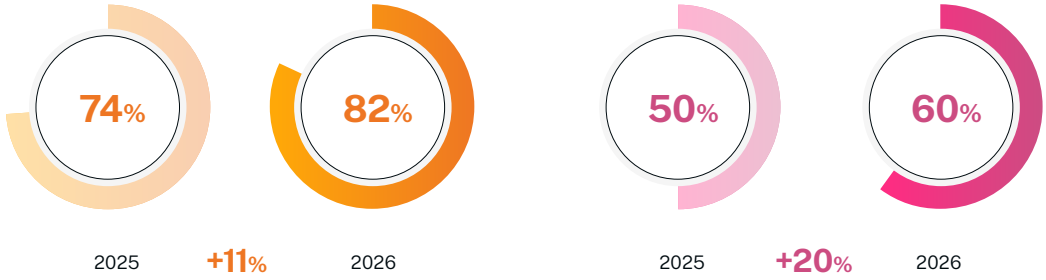
82% of organizations affected
(up from 74%, an +11% YoY increase)

Critical Security Debt:

60% of organizations affected
(up from 50%, a +20% YoY increase)

FIGURE 1

Prevalence of security debt and critical debt and among organizations



Application-Level Debt:

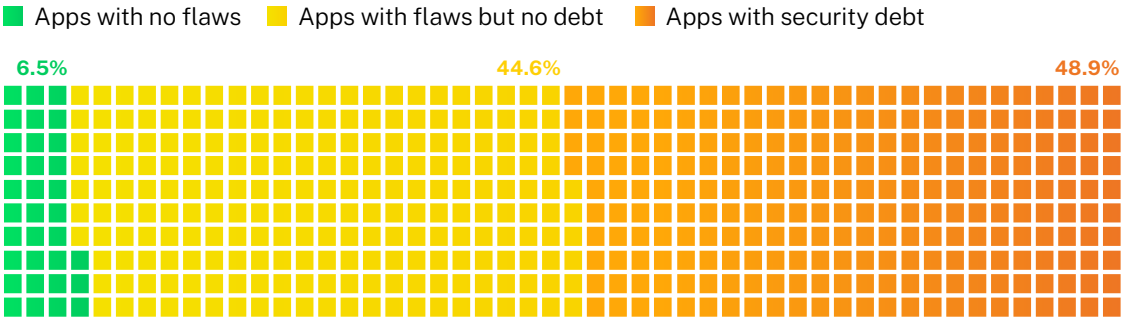
49% of apps now carry security debt
(up from 42%, a +17% YoY increase)

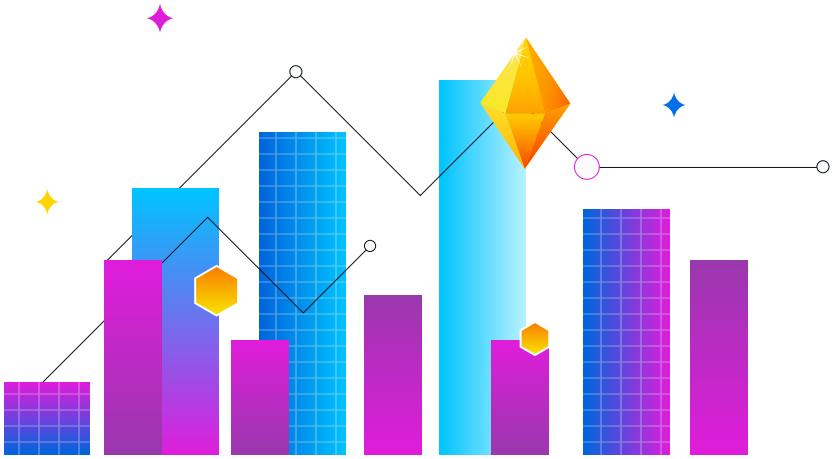
Apps with No Flaws:

Increased slightly to **6.5%** from 6.1%

FIGURE 2

Prevalence of security debt across all applications active for at least one year





Effective prioritization allows teams to focus on remediating the most critical vulnerabilities first, ensuring risk is minimized even when resources are constrained.

The security debt crisis has reached an inflection point. While 2025 celebrated a decade of progress in reducing flaw prevalence and improving Open Web Application Security Project (OWASP) Top 10 pass rates in first-party code apps, 2026 confronts us with a sobering reality: the backlog is growing faster than remediation capacity can eliminate it. This represents more than a statistical blip. It signals a fundamental mismatch between the pace of software development, the complexity of modern applications, and the available resources for security remediation.

The story behind these numbers reveals three intersecting pressures. First, organizations are discovering more vulnerabilities as their testing programs mature and expand across Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA) modalities. Second, the accelerating pace of software releases we see in DevOps and CI/CD practices creates a continuous influx of new code before existing vulnerabilities can be addressed. Third, the growing technical complexity of applications, particularly those incorporating AI-generated code and extensive third-party dependencies, makes remediation more complex and resource-intensive.

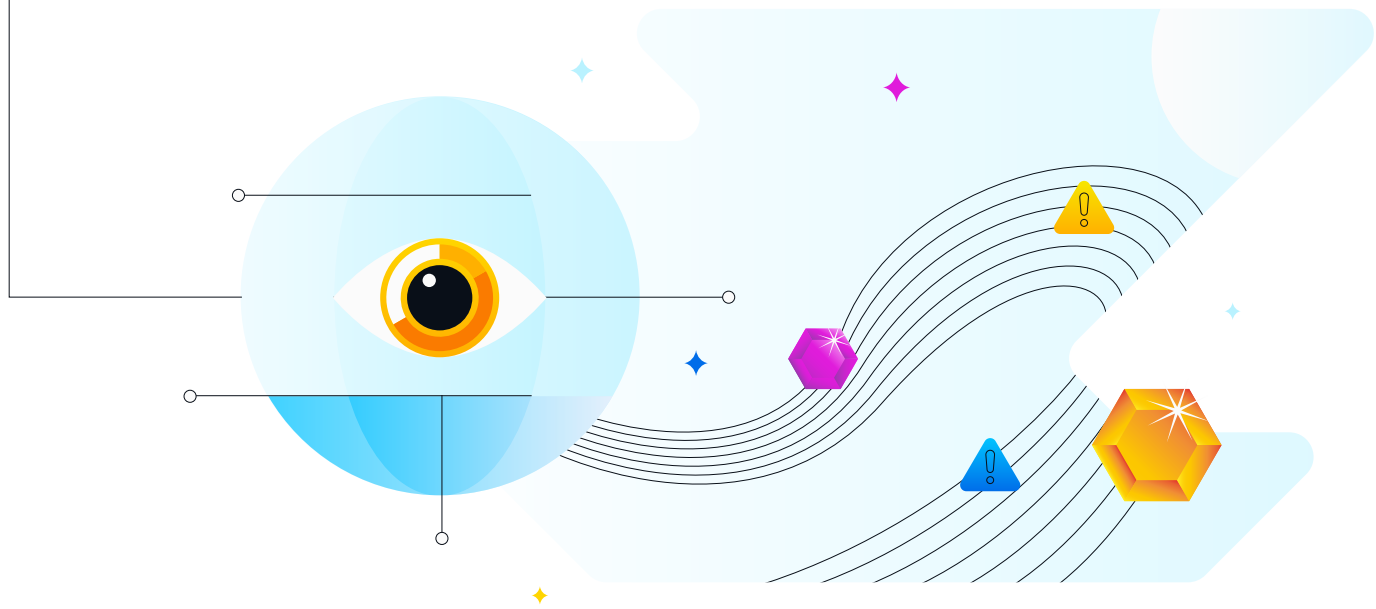
What makes this trend particularly concerning is the shift in critical security debt. The 20% relative increase in organizations carrying high-severity, long-unresolved flaws suggests that teams are increasingly forced to accept the risk or defer dangerous vulnerabilities, not because they're low priority, but because capacity constraints make comprehensive remediation too challenging for most organizations.

Later in the report, in Figure 6, we see a 4% YoY decrease in the half-life of flaws (meaning we're getting faster at fixing them), but it's still not enough to take on the number of flaws being introduced. This is why prioritization must become a core focus for modern development teams. Effective prioritization allows teams to focus on remediating the most critical vulnerabilities first, ensuring risk is minimized even when resources are constrained. By leveraging tools that provide context on vulnerability severity, exploitability, and business impact, organizations can make informed decisions about which issues to address immediately, and which can be scheduled for later remediation or a decision can be made not to remediate at all. This approach not only mitigates risk more effectively, it also helps maintain a balance between security and development velocity.

Understanding which applications constitute your organization's "crown jewels" is a critical component of effective prioritization. These are the systems and applications that hold the most significant value to your business – whether due to the sensitive data they process, their role in delivering core services, or their impact on overall operations. Prioritizing the remediation of critical security debt within these key assets ensures that your security efforts are focused where they matter most.

By leveraging AI-driven solutions, organizations can expedite the remediation process while streamlining prioritization efforts. AI not only helps identify vulnerabilities faster but also pinpoints the "crown jewels," enabling teams to focus their efforts where it matters most. Additionally, AI can provide actionable insights and automated fixes tailored to an application's environment and criticality, significantly reducing the manual workload for development and security teams. Beyond remediation, AI-driven solutions also play a pivotal role in managing compliance by continuously monitoring and aligning security practices with regulatory requirements. This holistic approach allows organizations to efficiently reinforce their most important defenses while ensuring adherence to compliance standards.

AI not only helps identify vulnerabilities faster but also pinpoints the "crown jewels," enabling teams to focus their efforts where it matters most.



CHAPTER 2

The High-Risk Vulnerability Surge

Vulnerabilities rated as both highly severe and highly exploitable—the 'high-risk region'—have increased by a relative 36%, representing a concerning concentration of dangerous, weaponizable flaws.

Key Point

High-Risk Region Growth:

36% relative increase in flaws with both high exploitability AND high severity (from 8.3% to 11.3%)

FIGURE 3

Breakdown of flaws according to severity and exploitability

Exploitability		Severity: LOW	MEDIUM	HIGH	VERY HIGH
		22.6%	57.8%	14.9%	4.6%
VERY LIKELY	22.2%	**0.9%**	**11.3%**	**8.4%**	**1.5%**
LIKELY	35.3%	**1.2%**	**32.6%**	**1.1%**	**0.3%**
NEUTRAL	33.8%	**13.7%**	**12.0%**	**5.3%**	**2.8%**
UNLIKELY	8.6%	**6.7%**	**1.8%**	**0.1%**	**0.0%**
VERY UNLIKELY	0.1%	**0.1%**	**0.0%**	**0.0%**	**0.0%**

High risk region totaling **11.3%**

The 36% surge in high-risk vulnerabilities since 2025 represents one of 2026's most critical findings, fundamentally challenging the narrative of steady security improvement. The concentration of flaws in the dangerous intersection of high severity and high exploitability has accelerated dramatically. There aren't just more vulnerabilities; there's more risk from vulnerabilities with real-world attack potential.

This trend likely reflects the convergence of several market forces. The proliferation of AI-assisted code generation tools may be introducing security flaws that many traditional scanning tools readily detect as high-severity issues. For example, a cross-site Scripting (XSS) attack can lead to severe data breaches, and in 86% of tests, AI-generated code failed security tests for XSS, according to the 2025 GenAI Code Security Report.

Meanwhile, the expanding attack surface created by microservices architectures, Application Programming Interface (API) proliferation, and cloud-native applications creates more opportunities for exploitation, even as organizations improve their basic security hygiene.

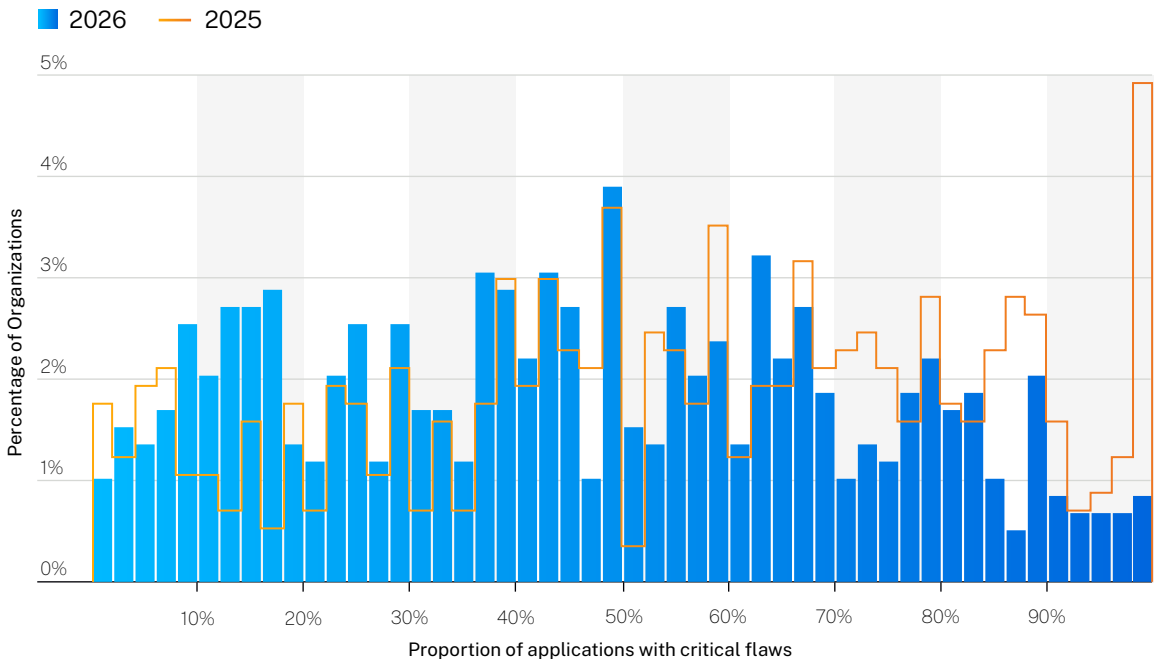
Proportion of Apps Affected by Critical Flaws

On the bright side, we see a favorable trend in the number of applications affected by these critical flaws. While there was a 36% relative increase in flaws with both high exploitability AND high severity, the proportion of applications affected by these critical flaws decreased. We see this in the density of the bars in Figure 4 shifting left from 2025 to 2026.

Fewer apps with high-severity flaws reiterates the importance of knowing which applications are your prized possessions and then prioritizing their protection within your security strategy. Identifying these critical assets allows teams to allocate resources effectively, focusing on remediation efforts that reduce organizational risk with the highest impact. Furthermore, incorporating real-time alerts and automated remediation processes can enhance responsiveness and ensure that vulnerabilities are addressed promptly.

FIGURE 4

High-severity flaw prevalence among organizations



CHAPTER 3

Modest Progress in Detection, Struggles in Remediation

Organizations continue to improve their ability to find vulnerabilities and introduce fewer flaws (overall flaw prevalence down from 80% to 78%), but remediation timelines show only incremental improvement and fix capacity remains constrained.

Key Points

Overall Flaw Prevalence in Apps Across All Scan Types:

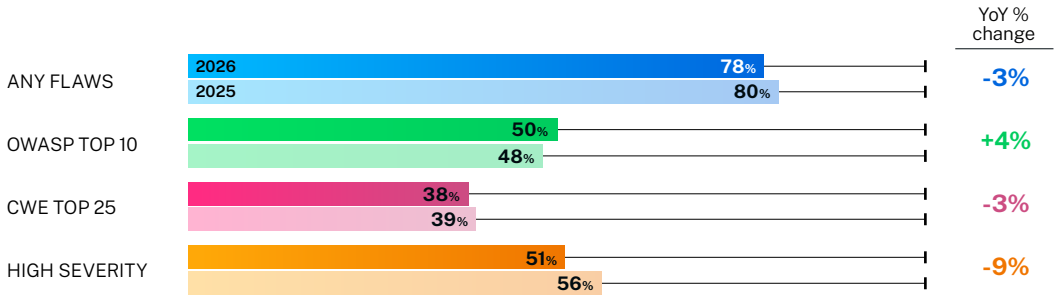
Decreased to **78%** (down from 80%, a -3% YoY improvement)

OWASP Top 10 Failure Rate:

Increased from 48% failing to **50%** failing (a +4% YoY increase in failing apps)

FIGURE 5

Percent of applications with security flaws across all scan types

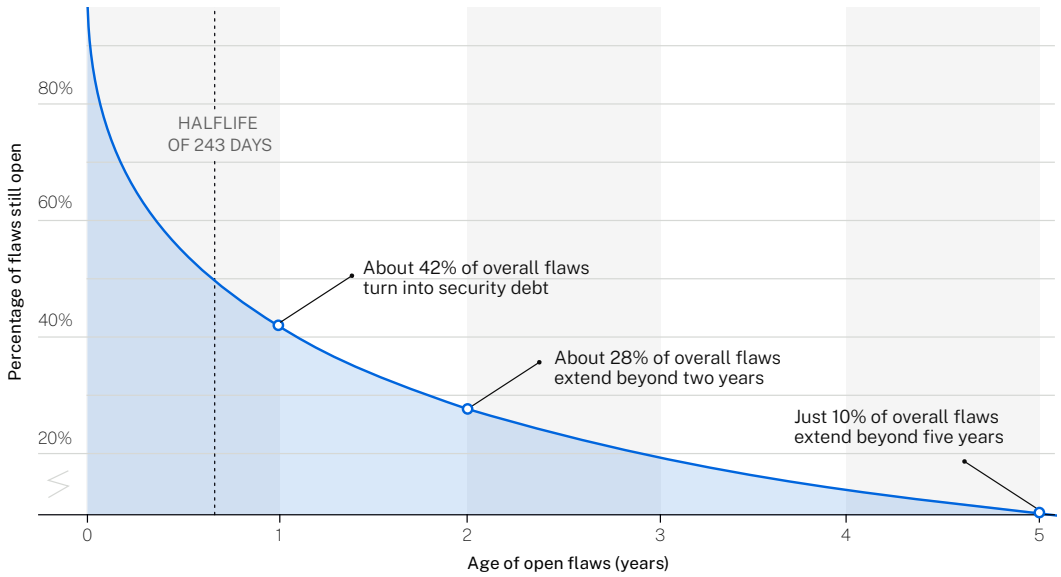


Fix Speed (Half-Life) Across All Scan Types:

243 days (down from 252 days, a -4% YoY improvement)

FIGURE 6

Overall flaw remediation timeline of all scan types based on survival analysis

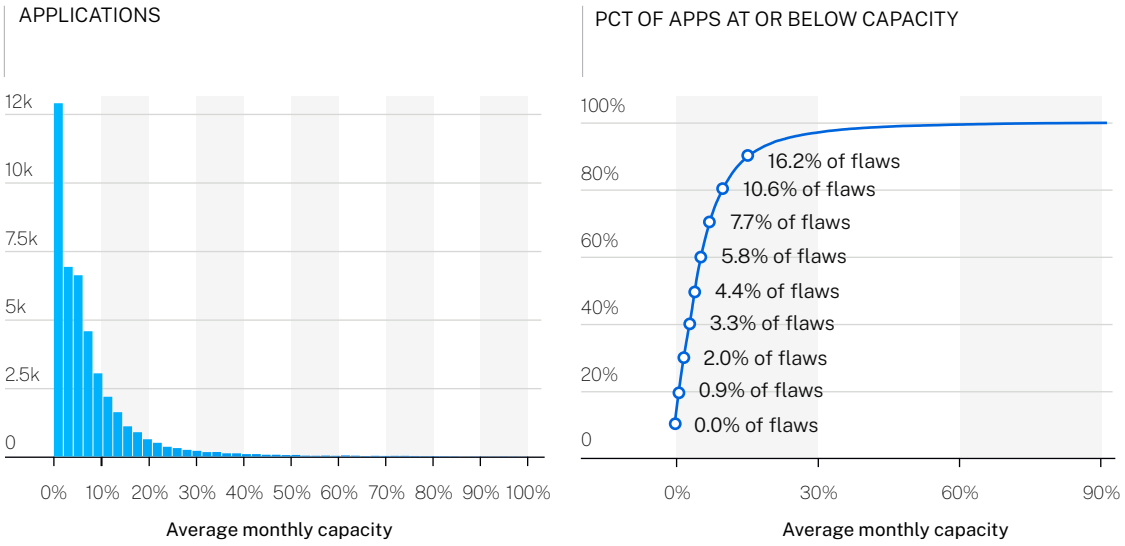


Monthly Fix Capacity:

Top performers at **16.2%** of flaws (down slightly from 16.5%)

FIGURE 7

Average monthly fix capacity across applications



The 2026 data tells a story of incremental progress shadowed by persistent structural challenges. On detection, organizations are winning: flaw prevalence across all scan types continues its multi-year downward trajectory and fix speed has improved by 4%, too. This reflects maturing AppSec programs, better secure coding practices, expanded use of automated testing, and likely the positive impact of security training programs.

Yet this improvement masks a profound prioritization and remediation failure. We see a 4% YoY increase in applications that have flaws ranked in the OWASP Top 10 as the 10 most critical risks to applications (Figure 5). And while fix speed improved (by just 9 days, from 252 to 243 days for half-life), this rate pales against the mounting volume of vulnerabilities discovered.

The math is unforgiving. If you're finding flaws faster than you're fixing them, and your fix capacity remains essentially flat (currently hovering around 10% of flaws monthly for median organizations), the backlog inevitably grows.

While we're optimistic that AI-generated fixes are a potential solution to reverse the tide, when we combine the high-severity flaw surge in Chapter 2 with the current rates of remediation, we can conclude that prioritization is key. Teams won't reduce more risk by getting faster at fixing the low-severity, unlikely-to-be-exploited flaws. Organizations need to focus on fixing the most critical and exploitable flaws in the most critical, "crown jewel" applications. And where does most of the critical debt reside—in first-party or third-party code? That brings us to the next chapter.

CHAPTER 4

The Persistent Third-Party Supply Chain Challenge

While third-party code shows slight improvement in overall security debt contribution, it continues to dominate critical security debt, representing 66% of the most dangerous, long-lived vulnerabilities. And since critical security debt rose a relative 20%, that slight decrease is a modest step in the right direction but not nearly enough to mitigate long-term risks associated with third-party code usage.

Key Points

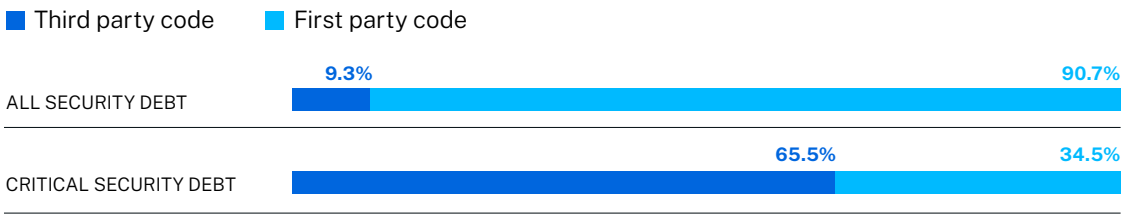
Third-Party Critical Debt:
66% (down from 70%, a -6% YoY improvement)

Third-Party All Security Debt:
9% (down from 11%, an -18% YoY improvement)

FIGURE 8

Proportion of security debt and critical debt in first-party vs. third-party code

Percentage of flaws



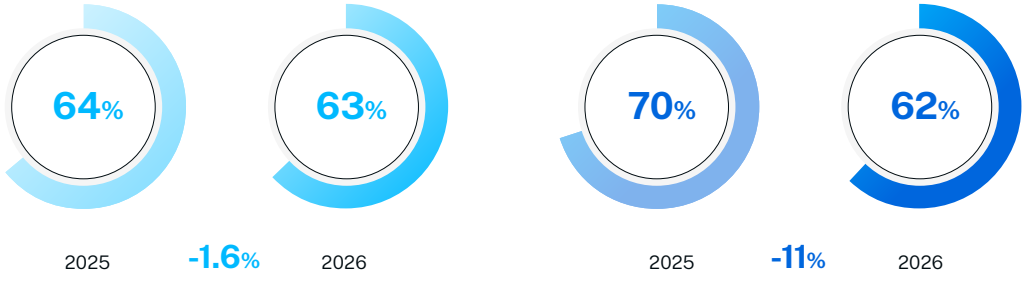
Applications with First-Party Flaws:
63% have flaws in first-party code (down from 64%, a -1.6% YoY improvement)

Applications with Third-Party Flaws:
62% have open-source vulnerabilities (down from 70%, a -11% YoY improvement)

FIGURE 9

Prevalence of flaws in first-party (left) vs. third-party (right) code among applications

Percentage of flaws

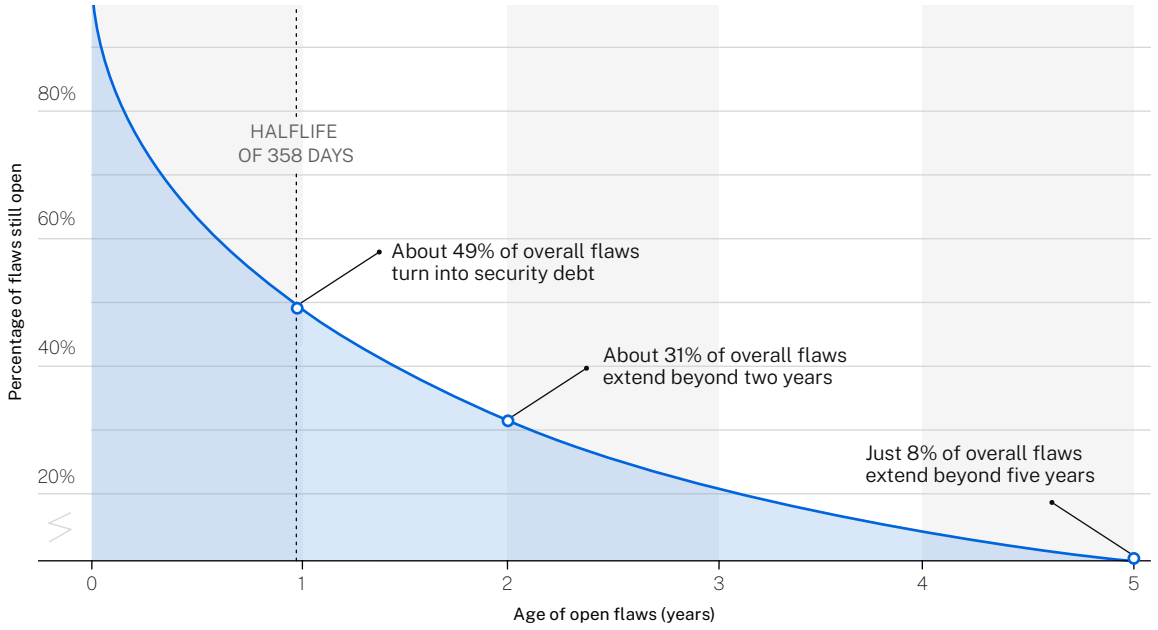


Remediation Half-Life for Third-Party Flaws:

358 days

FIGURE 10

Overall flaw remediation timeline based on survival analysis for SCA findings



The third-party security challenge presents one of 2026's most nuanced stories; it's a tale of modest improvement overshadowed by persistent structural concerns. The decline in third-party critical debt from 70% to 66% represents real progress, suggesting organizations are getting better at dependency hygiene, vulnerability scanning in the software supply chain, and potentially adopting package manager firewalls and Software Bill of Materials (SBOM) practices.

Additionally, the proportion of applications containing third-party vulnerabilities decreased by a relative 11%, falling from 70% to 62% of applications. We are pleased to report that this reduction indicates an improved security posture in open-source integration—especially on the application level, which indicates people may be taking supply chain action focused on their "crown jewel" applications.

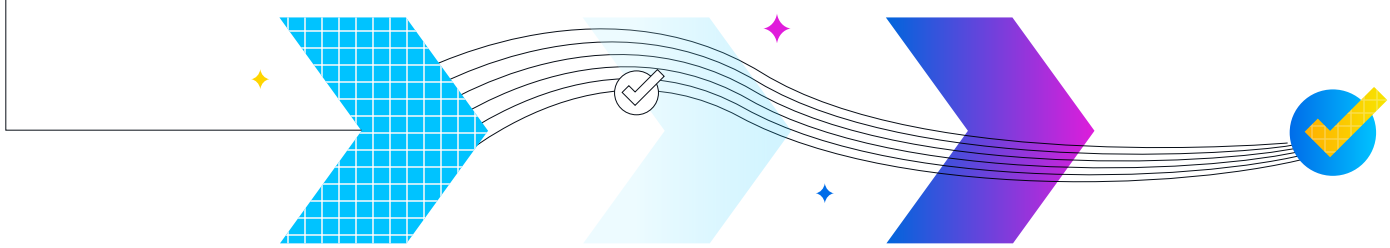
However, despite these gains, organizations still face significant challenges when addressing third-party code vulnerabilities. The half-life of third-party flaws found using SCA is 358 days (Figure 10); that's 115 days longer than the average of all scan types (Figure 6). That's almost four months longer!

One of the key hurdles to remediating vulnerabilities in the supply chain is the complexity of these chains. This complexity is further compounded by the nature of open-source flaws, which come in two types of dependencies: direct and transitive. Direct dependencies, where a configuration file references a library, are relatively straightforward to fix. However, transitive dependencies—where direct dependencies rely on other libraries—are far more challenging. Fixing transitive dependencies can risk breaking functionality in the direct library, often requiring code refactoring and significantly slowing the remediation process.

Reliance on open-source libraries and other third-party components introduces dependencies that, while accelerating development, often remain unchecked for vulnerabilities over time. This lack of visibility and accountability poses a significant challenge for organizations striving to maintain secure software. The difficulty of managing transitive dependencies exacerbates this issue, as teams must navigate the delicate balance between fixing vulnerabilities and preserving functionality.

To address this issue, organizations must prioritize strategies that include rigorous dependency management, frequent updates to third-party libraries, and proactive vulnerability scanning. Leveraging tools that integrate seamlessly into CI/CD pipelines can help automate this process, ensuring that outdated or vulnerable components are identified and remediated efficiently. Additionally, educating development teams on the importance of securing third-party code and adopting a "shift-left" approach to security can foster a culture of proactive risk mitigation.

Organizations must prioritize strategies that include rigorous dependency management, frequent updates to third-party libraries, and proactive vulnerability scanning. Leveraging tools that integrate seamlessly into CI/CD pipelines can help automate this process.



CHAPTER 5

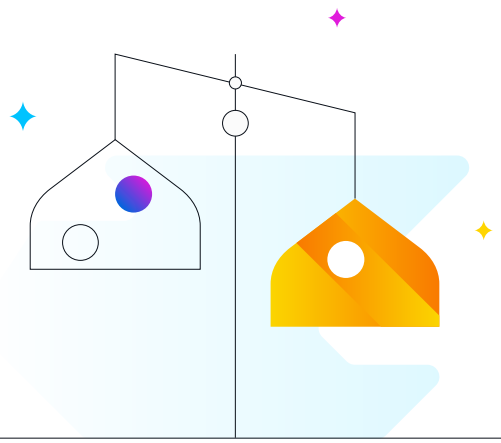
The AI Era's Double-Edged Impact

Though not explicitly measured in the data, it would be remiss not to discuss AI's expanding role in software development. AI presents us with a double-edged sword: simultaneously creating new vulnerability patterns and offering potential solutions for remediation at scale.

While it would be convenient for code generated by AI to be tagged as such, that's not the reality we live in. While the 2026 analysis doesn't provide definitive AI impact metrics, the data patterns and analyst questions reveal growing awareness that AI represents the wild card reshaping software security dynamics. The 2025 SoSS report noted that 'while many teams may not openly admit to using AI, other indicators of its presence and impact can be found'. 2026 trends suggest those indicators are becoming harder to ignore.

AI's rising integration into software development processes has redefined both the opportunities and challenges faced by AppSec teams. On one hand, AI-driven tools can enhance vulnerability detection, real-time code analysis, and even automated remediation workflows. These advancements enable organizations to address security risks faster than ever, improving team efficiency and reducing the likelihood of overlooked flaws. AI models can analyze vast amounts of code and identify patterns, allowing for a proactive approach in mitigating risks early within the software development lifecycle.

On the other hand, the rapid adoption of AI also introduces new attack vectors and raises concerns about the integrity of generated code. For instance, malicious actors can exploit vulnerabilities in AI-generated outputs or manipulate models through adversarial attacks. There's also risk from attackers finding your latent unfixed vulnerabilities with AI penetration tools. Furthermore, reliance on AI tooling without proper oversight may yield inaccurate results, such as false positives or missed threats, potentially undermining developer trust. To truly harness the potential of AI while mitigating risks, organizations must establish clear governance strategies, implement transparent model training processes, and ensure human oversight remains a core element of AI integration in security workflows.



Comparitive Analysis: Key Shifts from 2025 to 2026

Consistencies

What Stayed the Same and Why It Matters

1

The Fundamental Challenge Persists

78% of applications still contain flaws (78% in 2026 vs 80% in 2025), third-party code continues to drive critical security debt (over 65% range maintained), and remediation capacity remained on par.

Why It Matters:
Despite technological advances and increased awareness, the basic challenge of software security—ubiquitous vulnerabilities and constrained remediation—remains unchanged, leading to mounting security debt, validating the need for sustained focus.

2

Incremental Progress Continues on Detection

Flaw prevalence metrics continued trending positively year-over-year.

Why It Matters:
The multi-year pattern of detection improvement validates that secure coding practices, training, and automated testing are working—the problem is remediation capacity, not awareness.

Evolutions and Shifts

What Changed and What It Means

1

Security Debt Crisis Accelerates (+11% organizational prevalence)

From 74% of organizations with security debt (2025) to 82% of organizations with security debt (2026).

Driver: Detection outpacing remediation capacity; DevOps velocity; application complexity.
Implication: The remediation gap has reached crisis proportions; incremental improvements insufficient; transformational change required.

2

High-Risk Vulnerabilities Surge (+36%)

From 8.3% of vulnerabilities concentrated in the high-severity + high-exploitability category in 2025 to 11.3% in 2026.

Driver: Likely supply chain related; expanding attack surfaces; remediation not focused on severe flaws.
Implication: Prioritization frameworks must urgently shift to exploitability-weighted risk, not just severity; traditional CVSS scoring is insufficient.

3

Application Security Debt Grows (+17%)

From 42% of applications carrying security debt (2025) to 49% of applications carrying security debt (2026).

Driver: Accumulation outpacing elimination; long tail of hard-to-remediate flaws; resource constraints.
Implication: Nearly half of all applications are now burdened with year-old+ vulnerabilities, representing massive attack surface and technical debt.

Actionable Insights and Recommendations

For Organizations with High Security Debt (>50% of apps)

1

Emergency Triage Protocol

Immediately implement risk-based prioritization focusing on high-exploitability + high-severity intersection. Use ASPM solutions to correlate findings across tools and add runtime/business context.

Target:
Reduce critical security debt by 25% in 180 days through focused elimination of high-risk flaws.

2

AI-Assisted Remediation Pilot

Deploy AI remediation tools for the 'long tail' of simple, repetitive vulnerabilities. Allocate 10-15% of sprint capacity specifically to security debt reduction using AI-generated fixes with human review.

Target:
Increase monthly fix capacity from <5% to >10% within two quarters.

3

Dependency Management Overhaul

Implement package manager firewalls to prevent vulnerable dependencies entering the codebase. Establish dependency review process with security-weighted evaluation.

Target:
Reduce third-party critical debt contribution from 65%+ to <50% within one year.

For Organizations with Growing Application Portfolios

1

Shift Left on Remediation

Integrate automated fix suggestions directly into integrated development environments (IDEs) workflows. Implement 'fix before close' policies for high-risk vulnerabilities in new code.

Target:
Prevent net new security debt in a way that's prioritized; keep debt prevalence at <20% of applications.

2

Security Champion Enablement

Deploy security training focused on common vulnerability patterns in AI-generated code. Establish dedicated sprint points (15-20%) for security debt reduction.

Target:
Achieve fix half-lives of <90 days for critical vulnerabilities.

For Technology Leaders and Executives

1

Resource Reallocation

Recognize remediation capacity as a strategic constraint worthy of investment. Allocate budget for AI-assisted remediation tools, ASPM platforms, and security automation.

Target:
Double fix capacity through tooling investment, not just headcount.

2

Metrics and Accountability

Make security debt a board-level key performance indicator (KPI) alongside technical debt and SRE metrics. Tie security outcomes to development team Objectives and Key Results (OKRs) and performance reviews.

Target:
Organizational security debt is decreasing quarterly, with top-quartile performance expected within 18 months.

The Path Forward

The 2026 findings expose a glaring contradiction: improved detection masking a remediation crisis, incremental gains overshadowed by systemic debt accumulation, and pockets of excellence coexisting with widespread struggle. This isn't a story of failure. It's a story of transition, of an industry grappling with the implications of AI-accelerated development, cloud-native complexity, and the unforgiving mathematics of finite remediation capacity confronting an infinite attack surface.

The themes analyzed here—debt intensification, high-risk vulnerability surge, detection-remediation gap, persistent supply chain challenges, and AI's double-edged impact—form a tale of transformation imperative. Organizations have spent the last decade learning to find vulnerabilities. The next decade must be about learning to fix them in a prioritized way while simultaneously preventing their introduction in the first place.

The path is clear, the tools are available, and the examples exist. What remains to prevail in 2026 and beyond is to adopt the "Protect, Prioritize, Prove" strategy:

Prioritize

Find Clarity in the Chaos by Knowing What You Have & What Matters Most

The pursuit of fixing every flaw is a race that cannot be won. Instead, organizations must prioritize their efforts. This starts with identifying your "crown jewel applications": the applications and assets most critical to your business operations with the most material impact. By concentrating security resources on these critical areas and targeting the most severe, exploitable vulnerabilities, you maximize impact where it matters most. Prioritization is about visibility and risk ranking, because you can't protect everything equally.

Key questions to answer:

- What apps do I have?
- How many are there and what do they do?
- Which are public-facing vs. internal?
- Which handle sensitive data, models, or IP?
- What's my AI attack surface?

Protect

Enable Automation and Embrace DevSecOps to Secure Apps Continuously

With priorities in place, protection must become a strategic, continuous process focused on active risk reduction. Automation and DevSecOps practices are key to building scalable, efficient defenses. By integrating security into the development lifecycle and automating vulnerability detection and response, you reduce human error and keep pace with the rapidly evolving threat landscape. This ensures your critical applications remain secure as they grow, adapt, and interact within an ever-expanding software supply chain.

Key questions to answer:

- What's in my apps? How do I automate knowing this continuously?
- Where are the vulnerabilities—in models, data, prompts, pipelines, dependencies?
- How do I fix the risks in there at scale?

Prove

Software Assurance, Compliance, & Due Diligence

To prevail is to move beyond reacting to threats and toward assuring that your software and systems operate within a consistently compliant environment. Proving is not just about preventing recurrence; it's about demonstrating, with evidence, that your organization adheres to recognized security frameworks and regulatory requirements. Software assurance provides the foundation for this proof by ensuring controls are designed, implemented, and continuously validated across the development and operational lifecycle. A mature security posture makes compliance measurable, repeatable, and defensible. By aligning security practices to established frameworks and regulatory mandates, organizations can clearly demonstrate reasonable care, audit readiness, and operational discipline.

Transparency becomes a strategic advantage: verifiable evidence of compliance reassures regulators, customers, and stakeholders that security is not ad hoc, but systematically governed and enforced.

Key questions to answer:

- How do I ensure my software and environments continuously meet required security frameworks and regulatory standards?
- How can I provide clear, auditable evidence that compliance and assurance controls are operating effectively?

Taking control of your security landscape begins with a structured approach that empowers teams and builds resilient systems. By embedding proactive security measures into your workflows, you not only meet compliance demands but also create a foundation for innovation and trust. It's time to turn challenges into opportunities with a solution tailored to your organization's evolving needs.

Start your security transformation today. Schedule a demo to discover why organizations trust Veracode to reduce vulnerabilities, enhance efficiency, and achieve compliance with confidence.

Methodology

The report contains findings about applications that were subjected to static analysis, dynamic analysis, software composition analysis, and/or manual penetration testing through Veracode's cloud-based platform. Specifically, the data in this year's report comes from:

- 1.6M unique applications with 141.3M raw findings
  - 115.6M raw static findings
  - 3.6M raw dynamic findings
  - 22.1M raw SCA findings

This data represents companies of all sizes, commercial software suppliers, software outsourcers, and open-source projects. In most analyses, an application was counted only once, even if it was submitted multiple times as vulnerabilities were remediated and new versions were uploaded. For software composition analysis, each application is examined for third-party library information and dependencies. These are generally collected through the application's build system. Any library dependencies are checked against a database of known flaws.

The OWASP Top 10 was updated in November 2025 with minor changes. We used the prior version (2021) of the OWASP Top 10 for this analysis.

A Note on Mass Closures

While preparing the data for our analysis, we noticed several large single-day closure events. While it's not strange for a scan to discover that dozens, or even hundreds, of findings have been fixed (50% of scans closed fewer than 2 findings), we did find it strange to see some applications closing thousands of findings in a single scan. Upon further exploration, we found many of these to be invalid. These large collections of flaws were both added and removed in single scans: Developers would scan entire filesystems, invalid branches, or previous branches, and when they would rescan the valid code, every finding not found again would be marked as "fixed."

These mistakes had a large effect: The top 0.01% accounted for over 1 out of 10 of all the closed findings. These "mass closure" events have significant effects on measuring flaw persistence and time to remediation and were ultimately excluded from the analysis.

Appendix

This report would not have been possible without the invaluable contributions of several individuals and organizations. We extend our deepest gratitude to:

- **David Severski** and **Wade Baker** of the Cyentia Institute for their exceptional expertise in data analysis and statistical modeling, which provided the foundation for the insights presented in this report.
- **Natalie Tischler** and **Joe Ariganello** for their outstanding efforts in authoring and shaping the narrative of this report.
- **Karen Buffo**, **Niels Tanis**, **Chris Wysopal**, **Jens Wessling**, **Sohail Iqbal**, and **Katy Gwilliam** for their ideation on the right questions to ask, meticulous review and thoughtful feedback, ensuring the accuracy, clarity, and impact of the findings.

Your dedication and expertise have been instrumental in delivering the 2026 State of Software Security report. Thank you for your contributions to advancing the field of application security.

About Veracode

Veracode is a global leader in Application Risk Management for the AI era. Powered by trillions of lines of code scans and a proprietary AI-assisted remediation engine, the Veracode platform is trusted by organizations worldwide to build and maintain secure software from code creation to cloud deployment. Thousands of the world's leading development and security teams use Veracode every second of every day to get accurate, actionable visibility of exploitable risk, achieve real-time vulnerability remediation, and reduce their security debt at scale. Veracode is a multi-award-winning company offering capabilities to secure the entire software development life cycle, including Veracode Fix, Static Analysis, Dynamic Analysis, Software Composition Analysis, Container Security, Application Security Posture Management, Malicious Package Detection, Package Manager, and Penetration Testing.

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

VERACODE