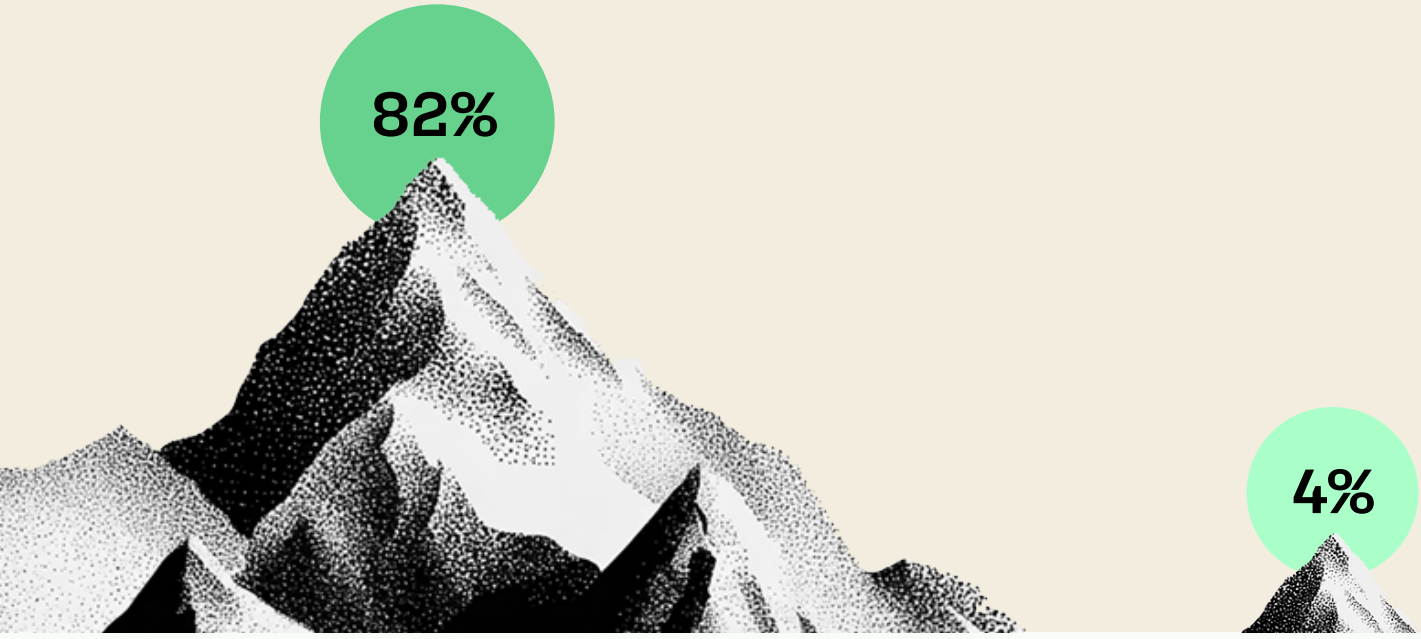
root

2026 Shift Out Benchmark Report

The Shift-Left Lie: Why 82% Claim Success While Only 4% Achieve It

A Root Research Report

Survey of 160 Senior Cybersecurity Decision-Makers | December 2025



“Claim Success”

“Achieve Zero Debt”

Executive Summary

We surveyed 160 cybersecurity decision-makers to answer one question: **Is shift-left security actually working?**

The answer reveals both a crisis and an opportunity. Organizations face a fundamental disconnect between perception and reality in vulnerability remediation, but emerging autonomous remediation technology can finally close the gap.

The Open Source Vulnerability Crisis

Modern applications are built on open source: container base images, language runtimes, and thousands of third-party dependencies and package managers like npm, pip, maven, gradle. Each brings known vulnerabilities (CVEs) that attackers actively exploit. For one-third of weaponized vulnerabilities, exploitation now occurs on or before disclosure day, leaving organizations with effectively zero remediation window.[3]

The pressure to remediate quickly creates its own risk. In November 2025, the Shai-Hulud attack compromised 700+ npm packages (including Zapier, PostHog, and Postman), turning "upgrade to latest" into an infection vector. For organizations whose only remediation path is upgrading, the trap is now complete.

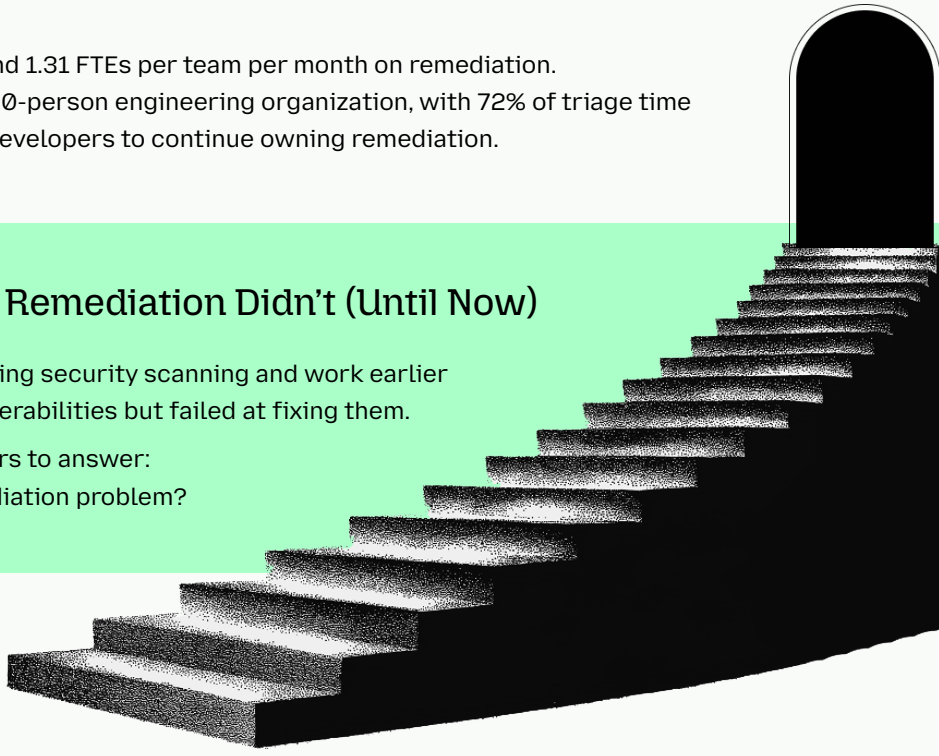
The business impact is measurable and severe:

- **Security Risk:** 33% of engineering and platform teams have 25% or more of their production containers running unpatched High or Critical CVEs older than 30 days.
- **Compliance Exposure:** 49% of security and GRC leaders cite compliance requirements such as PCI DSS, SOC 2, and ISO 27001 as their top urgency driver because unpatched vulnerabilities with known fixes are indefensible during audits.
- **Velocity Tax:** 60% of engineering organizations have experienced multiple release delays due to security findings, and 45% of teams still release with known High or Critical CVEs because slowing delivery is not an option.
- **Productivity Drain:** Engineering teams spend 1.31 FTEs per team per month on remediation. This costs $2.7M to $3.3M per year for a 100-person engineering organization, with 72% of triage time wasted on false positives. Only 12% want developers to continue owning remediation.

The Problem: Detection Scaled, Remediation Didn't (Until Now)

For a decade, the industry "shifted left" by moving security scanning and work earlier in development. This succeeded at finding vulnerabilities but failed at fixing them.

We surveyed 160 cybersecurity decision-makers to answer:
Is shift-left solving the open source CVE remediation problem?



The Perception Gap

82% of organizations believe their shift-left strategy has been “highly successful.”[1]

Yet only 4% have achieved zero CVE debt.

What They Belive	What The Data Shows
Shift-left is working	Only **4%** have zero CVE debt[1]
We’re managing vulnerabilities	**33%** have >25% of production containers with unpatched High/Critical CVEs older than 30 days
Our process is effective	Only **6%** say “our process works well”

82%
“Claim Success”

VS

4%
“Achieve Zero Debt”

This 78-point gap between belief and reality is our central finding.

Organizations aren’t failing because they lack tools or commitment. They’re failing because shift-left optimized for detection while ignoring remediation capacity. Detection scaled with automation. Remediation remained manual, scaling only with headcount.

The result:
Organizations detect thousands of vulnerabilities monthly but can fix only dozens. With MTTR measured in weeks while exploitation windows shrink to hours, this capacity mismatch has become a critical business risk.

Nobody noticed until the debt became unmanageable, teams burned out, and compliance audits started asking harder questions about those “known vulnerabilities with available fixes” still running in production.

But the solution has arrived. Autonomous remediation agents can now match detection’s scale by handling the full fix cycle (patching, testing, validation) with compute rather than headcount. Organizations deploying these capabilities are achieving zero-CVE outcomes while redirecting engineering capacity back to product development. The 4% who’ve solved this problem show it’s possible. The technology to join them exists today.

The Evidence: Six Symptoms of a Broken Model

The alternative: While manual remediation scales with headcount, autonomous remediation scales with compute, fixing vulnerabilities at the same pace they're discovered.

1. CVE Debt Is Accumulating Despite Investment

Organizations invest significant resources in vulnerability remediation:

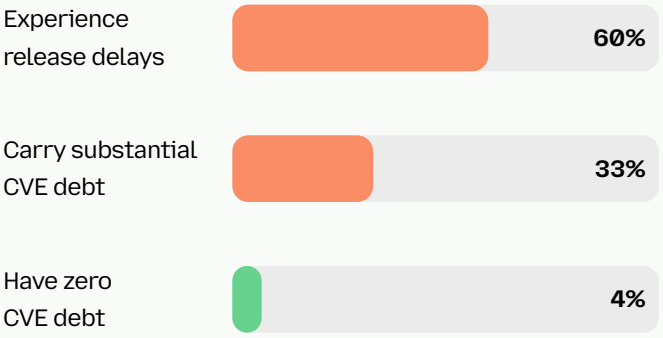
- **Average: 1.31 FTEs per team monthly** (210 hours) dedicated to remediation
- Larger enterprises (5,000+): **1.78 FTEs per team monthly**

Yet this investment fails to prevent debt accumulation:

- **33%** carry substantial CVE debt (>25% of production containers)
- **66%** routinely defer fixes because patching would break builds
- Only **6%** can remediate critical CVEs in under 24 hours

1.31

FTEs per team per month
~210 hours on remediation



The insight:

More resources won't solve a scaling problem. Manual remediation has hit its ceiling

2. Teams Are Burning Out

88% of organizations show observable signs of CVE-related burnout.

The most common manifestations:

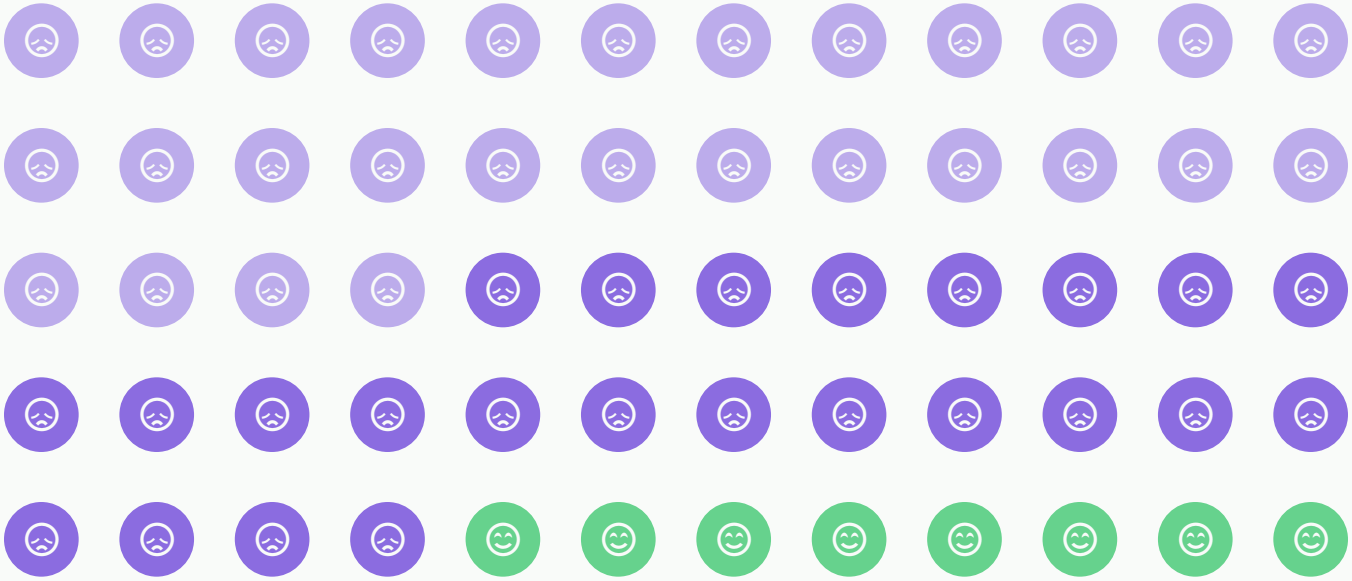
47%
Slower response to security incidents

45%
More errors or missed vulnerabilities

38%
Vocal frustration in meetings

Only **12%** report no burnout signs.

88% showing burnout signs, with " 47% slower incident response"



The insight:

Burnout isn't just a people problem; it's degrading security posture. Nearly half of organizations are responding more slowly to threats because their teams are exhausted.

3. Releases Are Being Delayed or Shipped Vulnerable

60% experienced multiple release delays due to security findings in the past year.

When delays aren't acceptable, organizations accept risk instead:

45% release with known High/Critical CVEs "sometimes" or more frequently

60%
with multiple delays

OR

45%
shipping with known CVEs

Ship late

Ship vulnerable

The insight:

The slowdown isn't just security itself; it's the manual, fragmented work required to meet security standards. Teams must choose between waiting on fixes or shipping with known risk. Neither is acceptable, and both show how the burden of "doing the right thing" has become a tax on delivery.

4. The Challenges Are Structural, Not Tactical

When asked about their DevSecOps experience, organizations cited systemic issues:

The Friction Problem

48%

Security findings create friction between teams

33%

Security is perceived as a delivery blocker

34%

Issues require leadership escalation

The Scale Problem

48%

Manual patching creates overwhelming workload

41%

Developers lack context to fix within SLAs

39%

Too many false positives drain capacity

External research validates the cost: Aikido Security found that developers spend **6 hours per week** triaging security alerts, with **72% wasted on false positives.** That amounts to **$20,000 per developer annually** in lost productivity, or **$20M per year for a 1,000-developer organization.**[2]

The Breaking Change Problem

33%

Breaking changes slow or prevent CVE resolution

66%

Routinely defer fixes due to breaking changes

63%

Rank application dependencies as #1 most painful layer

The insight:

These aren't problems you can train or hire your way out of. They're structural limitations of manual, developer-owned remediation.

5. The Developer Capability Paradox

Here's the counterintuitive finding:

65% agree their developers have the knowledge and capability to fix vulnerabilities.

Yet among those same organizations,
67% still defer critical fixes due to breaking changes.

The insight:

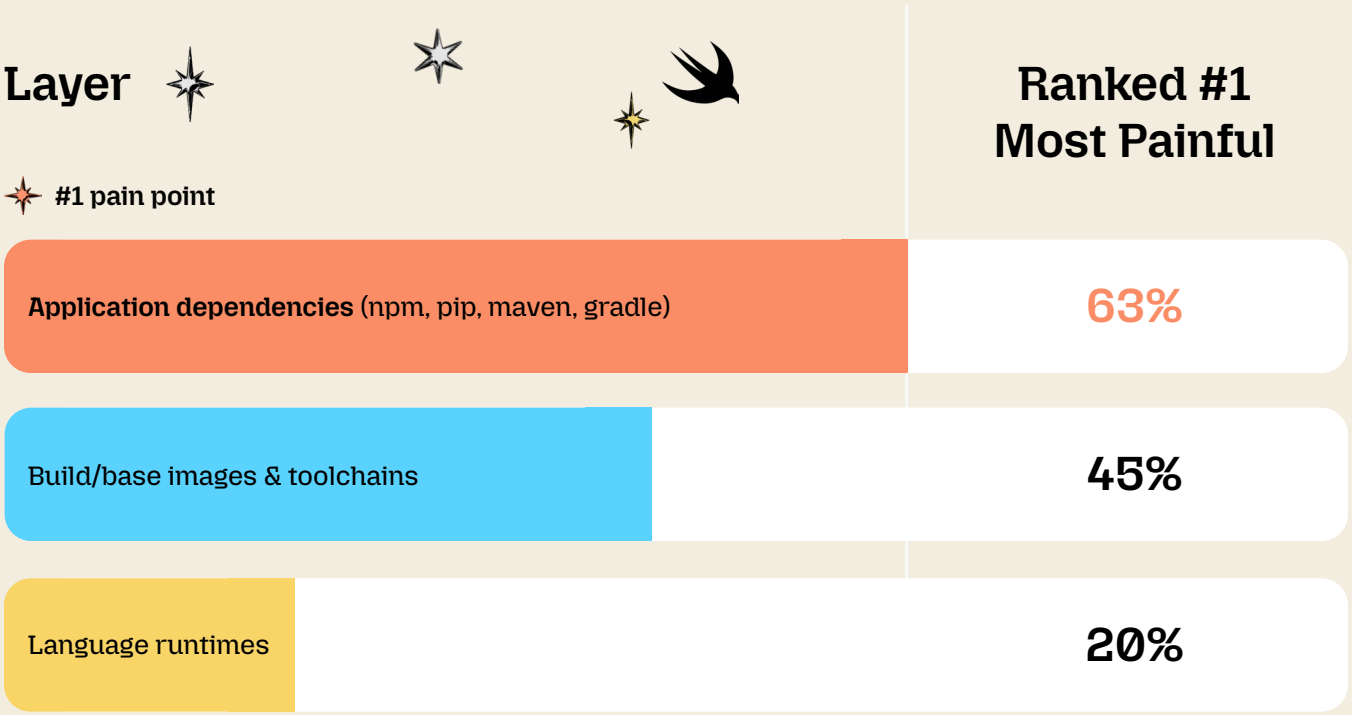
Knowledge isn't the bottleneck. Developers know how to fix vulnerabilities; they lack the time, priority, and incentive to fix them all without breaking things. Security work competes with feature development, and when vulnerability fixes require navigating complex dependency chains and testing for breaking changes, they lose. The problem is scale, complexity, and competing priorities, not capability.

The solution:

Autonomous remediation that removes developers from the critical path entirely, handling vulnerability fixes, dependency updates, and breaking change testing without human intervention.

6. The #1 Pain Point Everyone Ignores: Application Dependencies

When we asked which layers cause the most remediation pain, the answer was decisive:



THE MISDIRECTED INVESTMENT

While the industry debates base image security and invests in hardened container registries, 63% of practitioners say the real bottleneck is application dependencies, the layer vendors talk about least.

The industry obsesses over base image security. The actual bottleneck is application dependencies.

This is the layer where:

- Developers interact most frequently
- Version-pinning is strongest (“don’t touch it, it works”)
- Breaking-change fear is highest
- Transitive dependencies multiply the problem exponentially

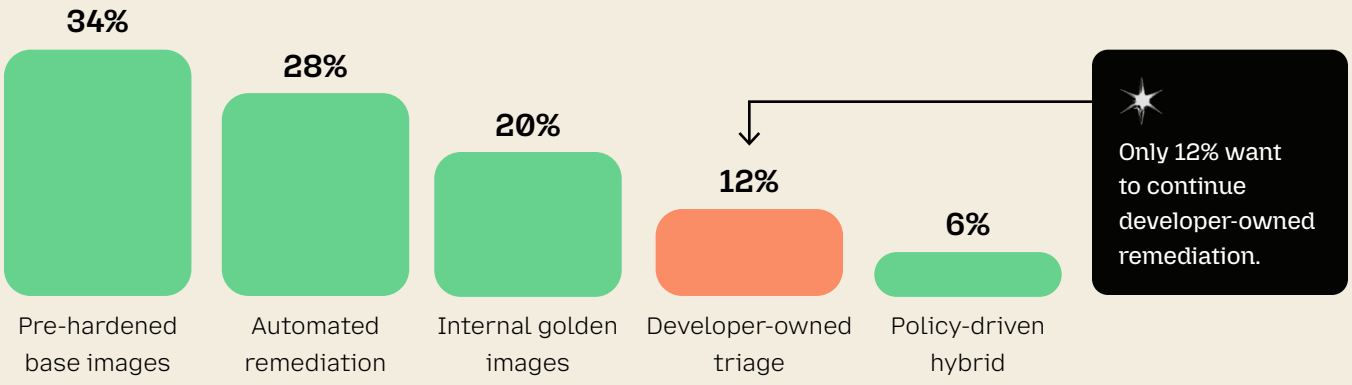
This is where the real toil lives and where automation delivers the biggest ROI.

The solution exists: automated remediation of npm, pip, maven, and gradle dependencies at your current pinned versions, eliminating both the toil and the breaking change risk that keeps 66% of organizations deferring fixes.

What Organizations Actually Want

We asked: "Which operating model would you choose to achieve zero-CVE containers while minimizing developer toil?"

Model	Support
Pre-hardened base images from trusted registry	34%
Automated remediation on existing images	28%
Language runtimes	20%
Developer-owned triage in sprints (status quo)	**12%**
Policy-driven hybrid	6%



The vast majority want someone, or something, else to handle it:

- **82%** prefer models that remove manual, per-vulnerability work from developers

When asked directly about automated remediation:

- **56%** are likely or very likely to adopt
- Only **9%** prefer manual approaches

The insight:

Organizations have learned that developer-owned remediation doesn't work. It's not just a scaling problem; it's fundamentally broken. It creates friction between teams, causes burnout, delays releases, and still accumulates debt. They're actively looking for alternatives.

What 82% want exists today: remediation handled outside developer workflows. Pre-hardened base images address the 34% preference, while automated remediation on existing dependencies addresses the 28% preference, covering the majority's desired operating models.

The Drivers: Why This Matters Now

What creates urgency to address container vulnerabilities?

External Pressure (Top 3)

49%
Industry compliance (PCI DSS, SOC2, ISO 27001)

48%
Known exploits actively used in the wild

47%
Cyber insurance or risk management requirements

Supply Chain Fear

53%
Concerned about zero-day vulnerabilities

52%
Concerned about malicious/compromised dependencies

The insight:

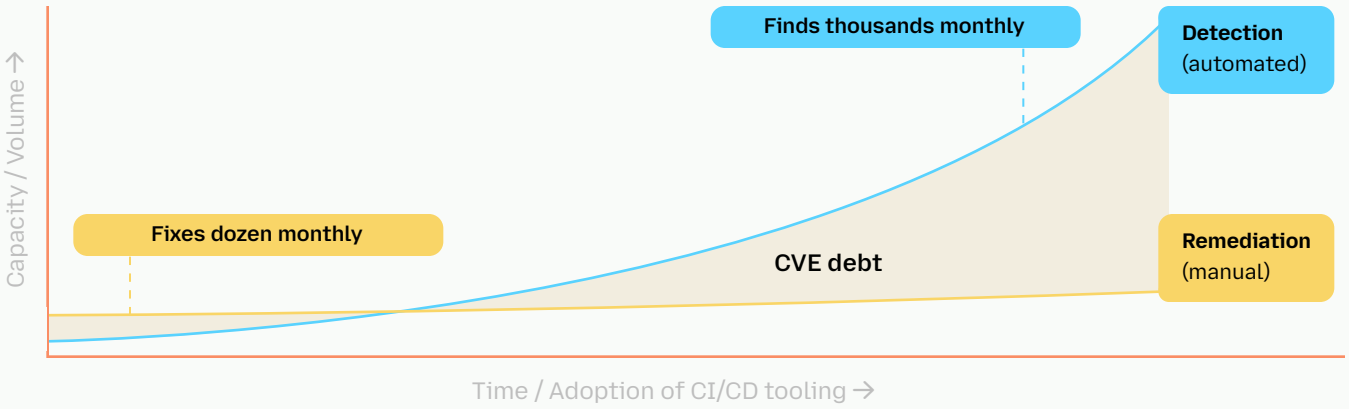
This isn't optional. Compliance, insurance, and active threats are forcing action and yet current approaches can't keep pace.

The Fundamental Problem

Detection scaled. Remediation didn't.

Detection	Remediation
Automated	Manual
Scales with compute	Scales with headcount
Runs on every build	Requires human judgment
Finds thousands monthly	Fixes dozens monthly

The result: **A growing gap between what organizations know about and what they can fix.**



THE WINDOW IS CLOSING: 32.1% of known exploited vulnerabilities (KEVs) show evidence of exploitation on or before the day the CVE is disclosed, up from 23.6% in 2024.[3] For one-third of weaponized vulnerabilities, the remediation window is effectively zero.

THE UPGRADE TRAP : In November 2025, the Shai-Hulud attack compromised 700+ npm packages (including Zapier, PostHog, Postman, and AsyncAPI), affecting projects with over 132 million monthly downloads. The malware executed during installation, before packages finished downloading. CISA issued an emergency advisory. For two days, "upgrade to latest" meant "install the worm." 53% of our respondents cited concern about malicious or compromised dependencies.

Shai-Hulud validated those fears. For organizations whose only remediation path is upgrading (and that describes most of them), there was no safe option. This exposes a gap the industry hasn't solved: remediation that doesn't require trusting that "latest" is safe.

THE SHIFT-LEFT ILLUSION: Shift-left moved security earlier in the pipeline. It did not solve the fundamental capacity problem; it simply revealed vulnerabilities faster than anyone could fix them. And with attackers increasingly exploiting vulnerabilities at disclosure or before, manual remediation is not just slow, it is structurally impossible for a growing share of threats.

The Path Forward: Autonomous Remediation

The gap between detection and remediation isn't permanent. Agentic AI is closing it.

Organizations are already seeing significant gains by deploying autonomous remediation agents that handle the full fix cycle (patching, testing, and validation) while keeping humans in the loop for oversight and exceptions. These agents scale with compute, not headcount, matching detection's automation with remediation's automation.

Our survey reveals clear demand signals:

88%
burned out on current approach

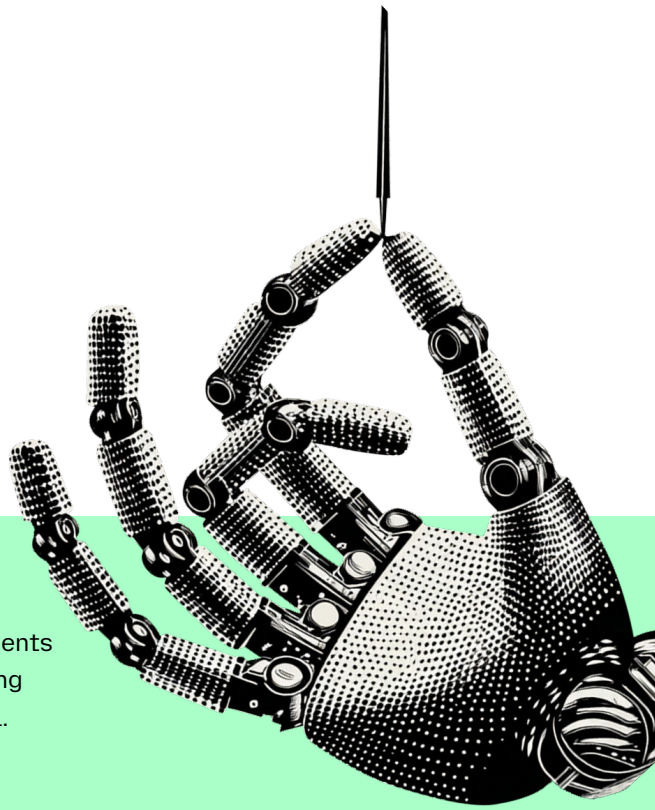
60%
experiencing release delays

56%
ready to adopt automation

12%
want to continue manual remediation

What organizations want:

1. Remove noise (address the false positive burden)
2. Avoid breaking changes (the #1 blocker)
3. Eliminate manual patching (the overwhelming workload)
4. Deliver zero-CVE outcomes (the goal 96% aren't achieving)



The technology to deliver this exists today. Autonomous remediation agents can fix vulnerabilities at your current pinned versions, eliminate breaking changes, and achieve zero-CVE outcomes without adding developer toil.

What This Means For You Next

For the CISO: You're Reporting Success While Accumulating Risk

The board problem you may not realize you have:
Your GRC dashboards likely show green. Shift-left is "implemented." Scanning coverage is 100%. Your team reports the program is working.

Our data suggests otherwise.
82% of your peers report shift-left success. Yet only 4% have achieved zero CVE debt. If your organization is like the majority, you are carrying unpatched High and Critical vulnerabilities in production, vulnerabilities with known fixes available, while reporting program success upward.

This is a material risk disclosure issue.

Finding	What It Means
45% release with known High/Critical CVEs	Risk acceptance has become routine, not exceptional. Are these decisions documented? Defensible?
47% report slower incident response due to burnout	Your team's ability to respond to the NEXT incident is degraded by fighting the current backlog
49% cite compliance as top urgency driver	When auditors or regulators look past your dashboards to actual production state, will the story hold?

The question to ask yourself:
If a breach occurs through a vulnerability that had a known fix available for 30+ days, what will your board presentation look like? "We knew about it, we had the fix, we didn't apply it" is not a defensible position and yet that's the reality for 33% of organizations in our survey.

The reframe:
Keep measuring program maturity but pair it with what actually reveals momentum: your **security debt trajectory.** Are you accumulating or reducing? If your 1.31 FTE monthly investment per team isn't shrinking the backlog, more investment in the same approach won't either.

The CISOs who will thrive are those who recognize that **remediation capacity, not detection capability, is now the constraint,** and invest accordingly.

For the VP of Engineering / Head of DevOps: Your Developers Have Voted

The productivity tax nobody budgeted for:

Our survey found teams invest **1.31 FTEs monthly,** 210 hours, on vulnerability remediation. That's **per team,** not organization-wide.

What this means at scale:

Org Size	Estimated Teams*	Monthly FTE on Remediation	Annual Hours Lost
50 engineers	7-8 teams	9-10 FTEs	18,000-20,000 hrs
100 engineers	14-17 teams	18-22 FTEs	36,000-44,000 hrs
500 engineers	70-85 teams	92-111 FTEs	184,000-222,000 hrs
1000 engineers	70-85 teams	183-223 FTEs	366,000-446,000 hrs

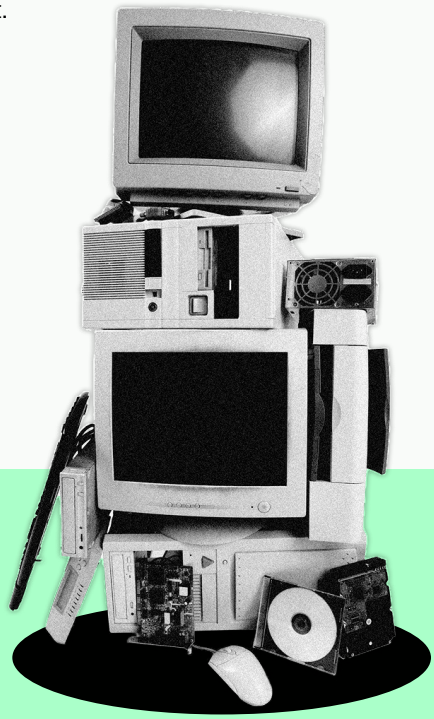
*Assuming 6-7 engineers per team/pod (industry standard for two-pizza teams)

For a 100-person engineering org, that's **18-22 full-time engineers' worth of monthly capacity** burned on triage, patching, testing, and coordination, not building product.

At $150K fully-loaded cost, that's **$2.7M-$3.3M annually** in remediation toil.

External research corroborates: Aikido Security found **15% of engineering time** is lost to triaging alerts alone, translating to **$20M annually for a 1,000-developer organization.**[2] And 72% of that time is wasted on false positives.

Yet it's not working: 60% still experience multiple release delays, and 33% are accumulating debtdespite the investment.



Here's what should really get your attention:

Only 12% want to continue developer-owned remediation in sprints.

Your developers, and your peers, have already concluded the current model doesn't work. They're not asking for more training or better processes. They're asking to be removed from the critical path.

Finding	The Developer Experience Reality
66% routinely defer fixes due to breaking changes	Developers aren't ignoring security—they're making rational trade-offs because fixes break things
65% have capable developers, yet 67% still defer	This isn't a skills gap. It's a time and complexity gap
48% say manual patching creates overwhelming workload	Your best engineers are spending cycles on toil instead of product
63% rank app dependencies as #1 pain point	The hardest problems are in code your team didn't write—npm, pip, maven—not base images

The velocity math:
Every sprint, your teams face a choice: ship features or fix vulnerabilities. With 60% experiencing multiple release delays from security, that choice is costing you competitive position.

Meanwhile, the 4% achieving zero CVE debt aren't doing it with more developers on security—they've automated remediation out of the developer workflow entirely.

The reframe:
At **1.31 FTEs per team,** you're essentially paying a full-time "remediation engineer" tax on every pod in your organization, except that tax is distributed across your best engineers as context-switching overhead rather than concentrated expertise.

The engineering leaders who win will be those who **treat remediation automation as platform infrastructure,** on par with CI/CD, observability, and testing, rather than asking product teams to absorb security toil indefinitely.

Recommendations

Accept the Scale Reality

Manual remediation at **1.31 FTEs per team** isn't preventing debt accumulation. Multiply that across your organization—more headcount won't solve a scaling problem.

Measure Outcomes, Not Inputs

Stop measuring tools deployed and vulnerabilities found.
Start measuring CVE debt trend and mean time to remediation.

Address Burnout as a Security Risk

88% showing burnout signs means 47% slower incident response.
This is a security posture problem, not just a people problem.

Evaluate Automation Seriously

56% are ready. Only 12% prefer the status quo.
The market has spoken, it wants automation.

Shift From Manual to Autonomous

The 4% achieving zero CVE debt have automated remediation out of developer workflows. Autonomous vulnerability remediation delivers this capability: continuous, automated fixes for both base images and application dependencies, with zero breaking changes and zero developer toil.

Conclusion

Container security has reached an inflection point.

82% believe shift-left is working. 4% have achieved zero CVE debt. That 78-point gap represents a decade of investment in detection without corresponding investment in remediation.

The evidence is overwhelming:

33%
carrying substantial CVE debt

88%
showing team burnout

60%
experiencing release delays

Only 12%
wanting to continue current approaches

This isn't a failure of execution. It's a failure of approach.

But here's the opportunity: autonomous remediation agents can now close the gap between what organizations detect and what they can fix. By automating the full remediation cycle with AI agents that scale with compute rather than headcount, organizations can achieve zero-CVE outcomes while redirecting engineering capacity back to product development.

Organizations that recognize this inflection point and adopt automation-first remediation will find competitive advantages: faster releases, healthier teams, stronger security, easier compliance.

Those that continue optimizing manual approaches will find the gap between requirements and capacity widening until something breaks.

The question isn't whether to change. It's when.

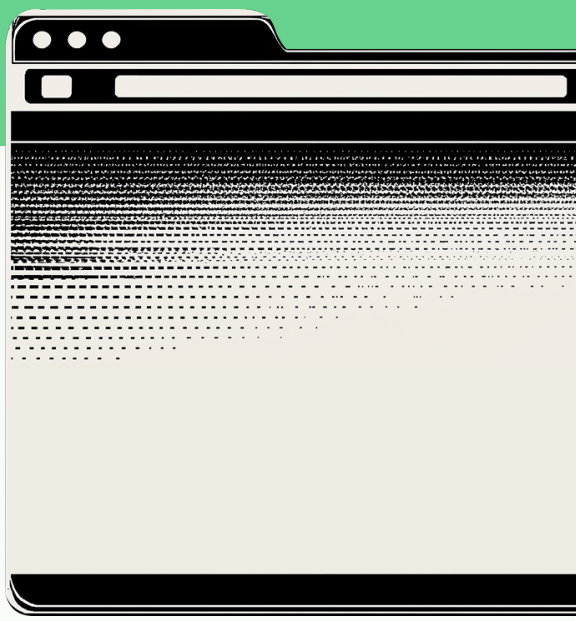
Stop shifting left. **Shift Out.**

Methodology

This vendor neutral, third party research was independently conducted by Virtual Intelligence Briefing (ViB). ViB's best in class market research design and analysis methodology delivers the industry's most accurate insights from precisely targeted, highly engaged members of ViB's community of more than ten million Technology Professionals, who are motivated to share their experiences and insights for the community's greater good. ViB leverages the market research industry's best practices and tools, incorporating extensive quality controls across the entire lifecycle from survey design to analysis and presentation of findings.

Survey: 160 cybersecurity decision-makers (November 2025)
Levels: Manager (39%), Director (23%), C-Level (14%), VP/SVP (13%), Staff (11%)
Company sizes: 50 to 10,000+ employees
Industries: Software (24%), IT & Services (23%), Healthcare (14%), FinTech (9%), Other (30%)
Functional areas: Engineering (75%), Architecture (60%), App Development (56%), DevOps (50%), AppSec (44%)

Sources



[1] Root & ViB Research, "State of Vulnerability Remediation Survey" (November 2025, n=160 cybersecurity decision-makers across enterprise organizations)

[2] Aikido Security & Sapio Research, "State of AI in Security & Development 2026" (n=450: 150 CISOs, 150 developers, 150 AppSec engineers)

[3] VulnCheck, "State of Exploitation: A Look Into 1H-2025 Vulnerability Exploitation" (July 2025)

root

The Shift Out Platform

Root is a comprehensive vulnerability remediation platform that eliminates the CVE grind by delivering open source that is clean of vulnerabilities, secured by default, and ready to use without engineering effort. Powered by thousands of specialized AI agents, Root's Agentic Vulnerability Remediation (AVR) system detects, patches, tests, and delivers fixed components in minutes with full transparency and no forced upgrades.

The platform enables AppSec teams to achieve instant remediation while engineers stay focused on building. Organizations close exposure windows by moving security at AI speed—across containers and libraries simultaneously.

Root Image Catalog (RIC)

Root's platform continuously remediates open-source base images covering 2,000+ containers across Python, Node, Java, Go, Ruby, PHP, Rust, .NET, and 40+ more. Drop-in replacements work seamlessly with your existing pipeline—no migration, no rebasing, no developer disruption.

Two SLA tiers: Standard (30-day Critical/High, 72-hour CISA KEV) and Enhanced (7-day Critical/High, 30-day Medium, 72-hour CISA KEV).

Root Library Catalog (RLC)

Root's platform secures open-source dependencies with continuous automated remediation across 8+ languages. Fix vulnerabilities at your current, pinned versions without forced upgrades or breaking changes. While fix-forward vendors force migration, Root's platform backports the smallest safe fix to your existing versions. SLA-backed fix rates provide predictable remediation capacity (1-25+ fixes/week).

Our Mission:

Secure all open source.