JupiterOne

CHECKLIST

Vulnerability Management: Can you answer these six questions?

A 5-minute self-assessment for security leaders who are tired of managing findings instead of reducing risk.

12,000+ high-severity findings a week. That's the median enterprise's queue in 2026 — and three or four tools are reporting the same CVE under different IDs.

Teams are drowning in data but still can't answer a basic question: which exposures represent real breach risk? More findings was never the problem. The problem is whether the exposures that matter are actually getting remediated.

So how do you know if your program is stuck in this trap? Answer the six questions below. If you're hesitating on more than one or two, you have a prioritization problem, not a scanning problem.

1

Can you measure how long your assets are vulnerable, and show that exposure time is decreasing?

☐ YES ☐ NO

If your program runs on a prioritized list of findings, you can probably show patch velocity and ticket close rate. But proving whether the exposures that represent actual breach risk are shrinking — that's a different story.

THE FIX →

Track continuous remediation state against your graph — not tickets or scanner counts. This lets you measure how exposure to crown-jewel assets changes over time, so you can demonstrate whether risk is going down.

2 Can your team turn 12,000 findings into what to remediate immediately?

☐ YES ☐ NO

The math is the message: prioritization that doesn't collapse findings into actionable work isn't prioritization. It's a queue. And queues grow until they become wallpaper.

THE FIX →

Narrow thousands of findings into a few dozen Remediation Plans, grouped by common fix (CPE-based) and routed by ownership, not category. Each plan carries the attack path that makes it critical and the fix that closes it.

3 Can you create one view across every scanner?

☐ YES ☐ NO

Each scanner assigns different identifiers, metadata, and severity ratings. Before remediation begins, someone has to determine which findings represent the same problem.

THE FIX →

Use cross-scanner deduplication to normalize findings from every scanner into one representation of each vulnerability, so every issue has one owner and one remediation plan.

4 Can you tell which of your thousands of findings form a viable attack path?

☐ YES ☐ NO

A 6.5 on an asset with a path to your production billing database is more dangerous than a 9.8 on a sandboxed dev environment. But CVSS alone doesn't tell that story.

THE FIX →

Use a graph that shows the blast radius to identify when a vulnerable asset sits on a path to a crown jewel. “Critical” becomes defensible to the audit committee, the carrier, the regulator, and the CFO, because it reflects your actual environment, not a theoretical score.

5 When someone outside security asks “Why this one first?”, does your answer hold up?

☐ YES ☐ NO

“We're patching the criticals” doesn't end the conversation anymore. Your audit committee wants trend data, not a status update. The carrier wants proof of control effectiveness before it renews. The regulator wants attestation it can file. And the CFO needs to know whether the exposure counts as a material risk to disclose.

THE FIX →

Compare findings against continuous technical data, mapped to compliance frameworks like SOC 2, DORA, and CIS. The same graph your security team uses to triage produces board-ready metrics, so the number in the boardroom traces back to the same evidence your analysts work from.

6 Do you know who owns the remediation?

☐ YES ☐ NO

Routing a finding to “the patching team” is a 20-year-old mistake. Platform ownership, SaaS admin, IAM lead, service developer — different exposures have different owners. If the case doesn't land with the person who can actually close it, in the system they already use, it doesn't move.

THE FIX →

Identify ownership during discovery, route remediation into existing workflows automatically, and verify that the exposure has been removed after the work is complete.

If you answered NO more than once or twice, you're not alone. Most programs get stuck the same way. Here's where.

Four mistakes to avoid

01

Don't rely on one input

Exploitability, business criticality, and reachability each tell part of the story. Lean on just one, and prioritization suffers.

02

Don't try to fix everything at once

Start with crown jewel exposure and expand outward.

03

Prepare the organization before changing workflows

Get buy-in from platform and dev leads before cases start landing in their queues.

04

Unify the data before replacing the tools

You don't need fewer scanners. You need one place that brings their findings together and produces a consistent view of exposure.