The DevSecOps Buyers Guide

How Security Leaders Can Embed Security into the Development Lifecycle Without Slowing Down Engineering

$20B+

DevSecOps market proejcted by 2030

93%

Of security leaders manage 3+ security tools

65 days

Average MTTR for critical vulnerabilities

DEFECTDOJO

Unify
Connect
Secure

Contents

01 Executive Summary

The DevSecOps market is projected to exceed $20 billion by 2030, growing at a CAGR above 13%, driven by regulatory mandates like Executive Order 14028, the EU Cyber Resilience Act, and the reality that breaches originating in application-layer vulnerabilities continue to dominate incident reports. Yet most organizations remain stuck in an early-maturity state: security teams manage an average of 3–6 disparate security tools, 71% report that security alerts contain significant noise from false positives and duplicates, and the average mean time to remediate a critical vulnerability remains 65 days.

DevSecOps promises to embed security into the software development lifecycle so that vulnerabilities are found earlier, prioritized by real risk, and remediated faster—without creating friction for developers. But the gap between aspiration and execution is wide. While 47% of organizations claim to regularly employ DevSecOps best practices, only 36% have actually adopted DevSecOps workflows, and tool sprawl remains the top barrier to effective implementation.

This guide provides security leaders with a structured evaluation framework for DevSecOps platforms. It covers the maturity spectrum from ad-hoc scanning to fully integrated security orchestration, details the ten capabilities every platform must have, compares architectural approaches, and provides a weighted evaluation scorecard. Throughout, we show how DefectDojo Pro addresses each requirement as a vendor-neutral DevSecOps orchestration platform with 200+ integrations, intelligent deduplication, risk-based prioritization, and native developer workflow support.

Key takeaway: DevSecOps is not a tool—it is an operating model that requires the right platform to orchestrate security across the entire SDLC. DefectDojo Pro provides that orchestration layer: 200+ tool integrations, intelligent deduplication to cut through alert noise, risk-based prioritization, and native push to GitHub, GitLab, Azure DevOps, Jira, and ServiceNow so developers fix vulnerabilities in the tools they already use.

02 What Is DevSecOps—and Why It Matters Now

Gartner defines DevSecOps as "the integration of security into emerging agile IT and DevOps development as seamlessly and as transparently as possible, ideally without reducing the agility or speed of developers or requiring them to leave their development toolchain environment." In practice, DevSecOps means that security testing, vulnerability management, and compliance checks are automated and embedded at every stage of the software development lifecycle—from code commit through production.

The shift from traditional application security to DevSecOps is driven by three forces:

Velocity has outpaced security

Modern engineering teams ship code multiple times per day. Traditional security review gates—manual pen tests, quarterly scans, release-blocking reviews—cannot keep pace. Security must move at the speed of development or it becomes a bottleneck that teams work around rather than through.

Tool sprawl creates noise, not signal

The average security team manages 3–6 security tools, and 57% of DevSecOps teams use five or more distinct tools. Each tool produces its own findings in its own format. Without a normalization and deduplication layer, teams drown in duplicates, false positives, and conflicting priorities. 71% of security leaders report that alerts contain significant noise.

Regulation now demands evidence of secure development

Executive Order 14028 (US), the EU Cyber Resilience Act, NIS2, and updated PCI-DSS requirements all mandate that organizations demonstrate secure-by-design development practices with auditable evidence. A DevSecOps platform must produce the compliance artifacts that regulators and auditors require.

Dimension	Traditional AppSec	DevSecOps
When security runs	End of cycle (release gate)	Every stage: commit, build, deploy, runtime
Who owns findings	Security team triages and assigns	Developers own findings in their tools (GitHub, GitLab, Jira)
Tool integration	Manual CSV exports, spreadsheets	API-first, automated CI/CD pipeline integration
Finding volume	Raw scanner output, high duplication	Deduplicated, normalized, risk-prioritized backlog
Remediation speed	65+ days average MTTR	11.5x faster with mature DevSecOps practices
Compliance evidence	Manual documentation	Automated audit trails and exportable reports

03 The DevSecOps Maturity Spectrum

Gartner's DevSecOps Maturity Model evaluates organizations across five dimensions: security skills and knowledge, developer enablement, secure design and threat assessment, security testing and scanning, and vulnerability management and remediation. Most organizations fall somewhere along a four-level maturity spectrum:

Level	Description	Characteristics	Key Gap
Level 1: Ad Hoc	Security testing happens sporadically, driven by audits or incidents. No consistent process.	Manual pen tests, spreadsheet tracking, no CI/CD integration, findings scattered across tools	No centralized vulnerability management or deduplication
Level 2: Foundational	Some security tools are in place but operate independently with limited automation.	SAST/DAST deployed but not in CI/CD, basic ticketing, findings reviewed manually	No normalization across tools; developer friction from context-switching
Level 3: Integrated	Security is embedded in CI/CD with automated scanning and centralized finding management.	CI/CD-integrated scanning, centralized platform, deduplication, SLA tracking, developer workflows	May lack risk-based prioritization or cross-domain coverage (AppSec + infra + SOC)
Level 4: Optimized	Risk-based orchestration across all security domains with automated remediation workflows.	Risk-based prioritization, rules engine, unified AppSec + SOC, MCP/AI integration, compliance automation	Continuous improvement—requires metrics-driven optimization of security posture

> ***Where DefectDojo Pro fits:*** *DefectDojo Pro is designed to move organizations from Level 2 to Level 4. It provides the centralized orchestration platform (Level 3) with the risk-based prioritization, rules engine, unified AppSec + SOC coverage, and AI integration that define Level 4 maturity.*

04 10 Capabilities Every DevSecOps Platform Must Have

1. Broad Security Tool Integration (150+ Parsers)

Your DevSecOps platform must ingest findings from every scanner in your stack—SAST, DAST, SCA, container, infrastructure, cloud, and custom tools—through native parsers, API connectors, or a universal import format like SARIF or CycloneDX. If your platform does not support a tool, you lose visibility.

2. CI/CD Pipeline Integration

Security scans must run automatically as part of every build—not as a separate process. The platform should offer CI plugins or API hooks for GitHub Actions, GitLab CI, Jenkins, Azure DevOps, and other pipeline tools, with the ability to set quality gates that block deployments when critical findings are detected.

3. Intelligent Deduplication

When five scanners report the same vulnerability, you need one finding—not five. The platform must normalize findings across tools and collapse duplicates using configurable matching algorithms. Without deduplication, alert noise makes DevSecOps unworkable.

4. Risk-Based Prioritization

Not all findings are equal. The platform must combine CVSS, KEV status, EPSS scores, asset criticality, exploitability, and business context to produce a risk-ranked remediation backlog. Developers should fix the most dangerous vulnerabilities first, not the loudest ones.

5. Developer Workflow Integration

Developers will not adopt a security platform that requires them to leave their IDE or project management tool. Findings must flow natively into GitHub Issues, GitLab Issues, Jira, Azure DevOps work items, or ServiceNow tickets—with full context, remediation guidance, and links back to the finding.

6. API-First Architecture

Everything the platform does must be automatable via API. Import findings, create engagements, query metrics, trigger scans, export reports. If a workflow requires clicking through a UI, it cannot scale with modern engineering velocity.

7. Rules Engine and Automation

Routine finding management should not require human effort. Auto-escalate critical findings, auto-assign based on product ownership, auto-tag by compliance scope, auto-close on verified remediation. A configurable rules engine eliminates repetitive triage work.

8. SLA Enforcement and Remediation Tracking

Define remediation SLAs by risk level, auto-notify owners before SLA breach, auto-escalate overdue findings, and track SLA compliance across products and teams. Without SLA enforcement, remediation timelines remain aspirational.

9. Compliance Reporting and Audit Evidence

The platform must produce exportable reports that satisfy auditors: what was scanned, what was found, how it was prioritized, when it was remediated, and what remains open. Support for frameworks like PCI-DSS, SOC 2, NIST, and CRA is essential.

10. Vendor-Neutral, No Lock-In

Your DevSecOps platform should work with any scanner, any CI/CD tool, and any ticketing system. If the platform only works well with one vendor's scanners or pushes you toward a single-vendor stack, you lose the flexibility to adopt best-of-breed tools as your program evolves.

---

05 DevSecOps Architecture Comparison

DevSecOps capabilities are packaged in fundamentally different architectures. Understanding these trade-offs is critical for selecting the right platform.

Category	Description	Strengths	Limitations
Open-Source + Commercial	Community-driven security platform with a commercial tier adding enterprise DevSecOps capabilities.	Transparent, vendor-neutral, no scanner lock-in, large community, 200+ integrations	Open-source tier may lack advanced automation; commercial tier needed for rules engine and risk prioritization
Scanner-Native Platforms	DevSecOps built into a scanning vendor's platform, tightly coupled with their own SAST/DAST/SCA tools.	Deep integration with vendor's scanners, single pane of glass for scan + manage	Lock-in to vendor's scanners, limited third-party tool support, biased prioritization
CI/CD Security Add-ons	Lightweight security plugins that bolt onto existing CI/CD platforms (e.g., GitHub Advanced Security).	Low friction for developers, native to existing pipeline, fast deployment	Limited to one CI/CD platform, shallow vulnerability lifecycle, no cross-tool deduplication
Developer Security Platforms	Developer-first security tools that embed scanning directly into the IDE, PR, and CLI workflow.	Low friction for developers, fast feedback loops, strong developer adoption	Typically limited to SAST/SCA, lack cross-tool deduplication, minimal lifecycle management (SLAs, compliance)

> ***Where DefectDojo Pro fits:*** *DefectDojo combines the open-source foundation (OWASP Flagship, 43M+ downloads, 200+ integrations) with a commercial tier that delivers full DevSecOps orchestration—CI/CD integration, intelligent deduplication, risk-based prioritization, rules engine, native developer workflow push (GitHub, GitLab, Azure DevOps, Jira, ServiceNow), and compliance reporting—without locking you into any scanner vendor. One platform from code commit through compliance.*

---

06 Evaluation Scorecard

Score each vendor 1–5 for each criterion, multiply by weight, and compare totals.

Capability	Weight	What to Evaluate	Vendor A	Vendor B	Vendor C
Tool Integration Breadth	**15%**	Number of native parsers, API/universal import, SARIF/CycloneDX support			
CI/CD Pipeline Integration	**10%**	Native plugins for GH Actions, GitLab CI, Jenkins; quality gate support			
Intelligent Deduplication	**15%**	Cross-tool dedup accuracy, configurable matching algorithms, noise reduction			
Risk-Based Prioritization	**15%**	Composite scoring (KEV, EPSS, asset context), configurability, transparency			
Developer Workflow Push	**10%**	Native push to GitHub, GitLab, Jira, Azure DevOps, ServiceNow with context			
API-First Architecture	**5%**	Full REST API, Python client, webhook support, automation coverage			
Rules Engine & Automation	**10%**	Auto-escalate, auto-assign, auto-tag, auto-close; no-code rule builder			
SLA Enforcement	**5%**	Risk-based SLAs, auto-notify, auto-escalate, compliance dashboards			
Compliance Reporting	**10%**	PCI-DSS, SOC 2, NIST, CRA support; exportable audit evidence			
Pricing & Vendor Neutrality	**5%**	Transparent pricing, no per-app/per-asset penalties, scanner independence			
TOTAL WEIGHTED SCORE	**100%**				

Tip: During your POC, import the same multi-scanner dataset into each vendor and compare deduplication rates and prioritized output. Ask: how many duplicate findings did the platform collapse? Do the top 10 priorities match what your team would choose manually?

---

07 Common Pitfalls to Avoid

Pitfall 1: Buying a scanner, not a platform

A SAST or DAST tool is not a DevSecOps platform. Scanners find vulnerabilities; platforms manage the entire lifecycle from discovery through remediation and compliance. If you buy a scanner expecting DevSecOps, you will end up building the orchestration layer yourself—or adding yet another tool.

Pitfall 2: Ignoring developer experience

If security findings land in a portal that developers never visit, they will not get fixed. The 45% of teams that report misaligned workflows are telling you that tools must meet developers where they already work—GitHub, GitLab, Jira, Azure DevOps. Developer adoption is the single biggest predictor of DevSecOps success.

Pitfall 3: Underestimating deduplication

Multiple scanners finding the same vulnerability should not create multiple work items. Without intelligent deduplication, your remediation backlog inflates by 3–5x, developers waste time on duplicates, and your metrics become meaningless. 71% of security leaders cite alert noise as a critical problem.

Pitfall 4: Locking into a single-vendor stack

Scanner vendors that offer "built-in DevSecOps" typically optimize for their own tools and provide limited support for third-party scanners. As your security program matures, you will want to adopt best-of-breed tools for each domain. A vendor-neutral platform preserves that flexibility.

Pitfall 5: Treating DevSecOps as a tool purchase

DevSecOps is an operating model, not a product category. The platform enables it, but success requires process changes, cultural alignment, developer training, and executive sponsorship. Organizations that treat DevSecOps as a tool deployment see minimal improvement. Plan for the change management alongside the technology.

Pitfall 6: Per-application or per-asset pricing

As your DevSecOps program scales—more applications, more scanners, more findings—your costs should not scale linearly. Per-app or per-asset pricing creates perverse incentives to limit security coverage. Demand pricing that rewards broader adoption.

---

08 Why DefectDojo Pro

DefectDojo Pro is the only DevSecOps platform that combines open-source transparency, 200+ native tool integrations, intelligent deduplication, risk-based prioritization, and native developer workflow push—all within a full vulnerability lifecycle management platform that includes SLAs, a rules engine, compliance reporting, and unified AppSec + SOC coverage.

What sets DefectDojo Pro apart for DevSecOps

- **200+ native integrations, zero lock-in.** DefectDojo Pro ingests findings from 200+ security tools—SAST, DAST, SCA, container, infrastructure, cloud, and SOC sources—through native parsers, API connectors, and a Universal Parser that accepts any data source. You are never locked into a single scanner vendor.
- **Intelligent deduplication that cuts noise.** When multiple scanners report the same vulnerability, DefectDojo's configurable deduplication engine collapses duplicates into a single finding, dramatically reducing the remediation backlog and eliminating the alert fatigue that undermines DevSecOps adoption.

- **Risk-based prioritization.** DefectDojo Pro's prioritization engine (launched May 2025) combines KEV status, EPSS probability, exploitability, reachability, revenue impact, and compliance exposure into a composite Priority score (0–1150) and Risk classification (Low → Urgent) for every finding.
- **Native developer workflow push.** Findings flow directly into GitHub Issues, GitLab Issues, Jira tickets, Azure DevOps work items, and ServiceNow tickets—with full vulnerability context, remediation guidance, and bidirectional sync. Developers fix vulnerabilities in the tools they already use.
- **API-first with CI/CD integration.** Full REST API with Python client library. Record security tests for each build with build ID, commit hash, branch, and source repo. CI plugins for GitHub Actions, GitLab CI, and Jenkins. Quality gates that block deployments when critical findings are detected.
- **Rules Engine for automated triage.** Auto-escalate critical findings, auto-assign based on product ownership, auto-tag by compliance scope, auto-close on verified remediation—all configurable without programming.
- **Unified AppSec + SOC.** The first platform to unify application security findings and SOC alerts under a single deduplication and prioritization model, eliminating the gap between development and security operations.
- **AI-first with MCP.** The first DevSecOps platform to support the Model Context Protocol, enabling AI models to access vulnerability data for intelligent triage and remediation guidance.
- **Predictable pricing.** No per-application, per-asset, or per-finding penalties. Your cost does not spike as coverage expands.

09 Next Steps: Your 30-Day Evaluation Plan

Adapt this framework to your procurement timeline.

Week	Activity	Key Deliverable
Week 1	Inventory your current security tools, CI/CD pipelines, and ticketing systems. Define DevSecOps requirements: which integrations are critical? What deduplication and prioritization capabilities do you need? Prepare multi-scanner test data.	Requirements doc, tool inventory, scorecard with custom weights, test data package

Week	Activity	Key Deliverable
Week 2	Vendor demos with your data. Import the same multi-scanner dataset and compare: deduplication rates, prioritized output, developer workflow integration, and CI/CD pipeline setup.	Completed scorecards; shortlist of 2–3 candidates
Week 3	POC: import production scan data, configure deduplication, set up risk-based prioritization, connect CI/CD pipelines, test developer push (Jira/GitHub/GitLab), and configure SLA policies.	POC results: dedup rate, prioritization accuracy, developer adoption feedback, SLA setup effort
Week 4	Evaluate pricing, negotiate terms, confirm deployment model and data residency. Build the business case quantifying noise reduction and MTTR improvement.	Vendor selection with business case and projected MTTR improvement

Ready to get started? DefectDojo Pro offers a free 2-week trial with full access to 200+ integrations, intelligent deduplication, risk-based prioritization, the rules engine, developer workflow push, and CI/CD pipeline integration. Visit defectdojo.com/get-started-with-pro to begin.

This guide was produced by DefectDojo. For questions, contact info@defectdojo.com or visit defectdojo.com.

Market data sourced from Grand View Research, Mordor Intelligence, Fortune Business Insights, Gartner DevSecOps Maturity Model, StrongDM DevSecOps Statistics 2025, Seemplicity 2025 Remediation Operations Report, and publicly available vendor documentation.