6 Pillars of a Successful **DevSecOps Practice**

splunk>
turn data into doing®

The meteoric growth of DevOps today is far from surprising. Among its many benefits, DevOps allows companies to shorten the software development life cycle (SDLC) and provide continuous delivery of high-quality software. As companies continue to transition to the cloud, DevOps helps them rapidly grow and evolve, accelerating time to market and boosting time to value for customers.

The easy availability of cloud computing resources and proliferation of open source software/code repositories have helped companies become prolific software producers. As a result, DevOps has become an alluring option for a larger number and wider variety of projects, boosting it from relative obscurity into what is now a widely accepted and mainstream set of practices. According to a 2021 survey of US and UK cloud purchase decision makers conducted by ClearPath Strategies, an independent strategic consulting and public opinion research firm, 62% of companies have standard DevOps practices, either across the organization or on a team-by-team basis. Another 28% have DevOps integrated into specific teams.

However, DevOps has also amplified security challenges by creating a larger, and more rapidly evolving attack surface — including many more points of potential compromise. And while organizations need to integrate security earlier and more comprehensively in the DevOps process to address these issues, traditional security practices and DevOps are clearly at odds. This conundrum has sparked a growing interest in security as a critical part of the DevOps practice, leading to the development of a new discipline — development security operations (DevSecOps).

DevSecOps ensures secure software delivery at the pace of DevOps. However, because DevSecOps is still an emerging discipline, many organizations are confused about what it means and requires, and are uncertain how they can adopt it effectively. In the following guide, we'll shine a light on how organizations of all sizes can take both a strategic and tactical view to create a sustainable DevSecOps practice. By using the six pillars outlined below, organizations can lay the foundation for a successful DevSecOps strategy and drive effective outcomes, faster.

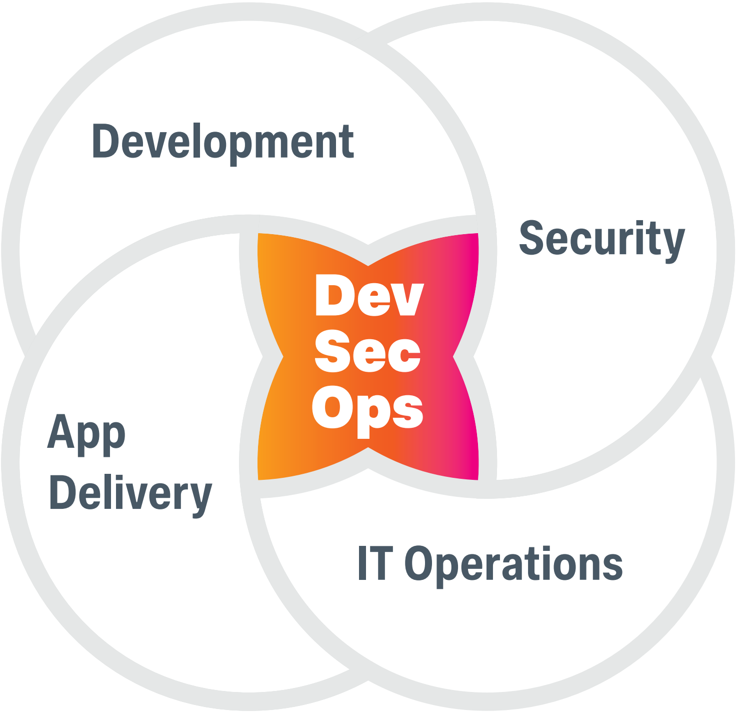
DevSecOps:
Shifting security to the left

Historically, responsibility for security belonged to a group of specialists who examined and stress-tested applications at the beginning and/or end of the development cycle. However, with the current pace of DevOps practices, traditional approaches that bolt on security as an afterthought aren't sustainable.

At its core, DevSecOps is about integrating security at every phase of the DevOps development lifecycle, from initial design and coding to testing, deployment and running. This allows practitioners to identify and remediate security vulnerabilities much earlier in the DevOps cycle, creating better-quality code and fewer fire drills in later stages.

To be effective, DevSecOps also requires all teams to share responsibility for the security of the application and its environment — with security, development and operations teams working toward shared goals. This may be easier said than done because each team is driven by different priorities: development teams are typically focused on speed and quality of code; operations teams on stability and resiliency of architecture; and security teams on thoroughness, broad coverage and assurance against weaknesses or vulnerabilities. A successful DevSecOps program brings all of these goals into alignment so organizations can create applications at either the same or a faster pace — with a lot more security built in.

Organizations also need to move faster to stay competitive, and developers may be hesitant adopters at first — either because they fear slowing down the pace of DevOps or because they lack experience with secure development practices. But when done right, DevSecOps is not only beneficial to the security team, it also helps developers increase productivity and on-time delivery of higher quality products. For the company, DevSecOps can reduce risk and deliver a superior security posture while also supporting the rapid pace of development.



6 pillars for DevSecOps success

While the benefits of DevSecOps are undeniable, successfully implementing DevSecOps is not a simple process. Essentially, DevSecOps is a practice, not a singular product, requiring a mindset and culture shift within an organization as much as new technological approaches and toolsets. This means bridging gaps between teams that have traditionally worked independently with their own tools and workflows, and often with different KPIs.

To be effective, a DevSecOps strategy needs to build on existing systems, removing obsolete processes or technologies, while also adding new ones as needed. In addition, it should support the needs of the development, operations and security teams, and cover all the layers of the technology stack as well as of the application itself, through all the stages of the SDLC.

Below are six pillars for creating and sustaining a successful DevSecOps strategy, and how Splunk can help facilitate and accelerate these approaches.

1 **Break down organizational silos:** To effectively implement DevSecOps, organizations first need to break down the barriers between development, operations, site reliability engineering (SRE) and security teams, and make security a shared responsibility. Teams should align on a common set of objectives and KPIs. While this will inevitably involve trade-offs for all involved, prioritizing and aligning on the most important goals without adding to the security debt will help foster adoption. Tools that enable users to work from a common source of truth are critical to supporting collaboration.

Splunk's approach: Splunk helps break down organizational silos and supports collaboration by providing a common platform and specialized solutions for security, IT and DevOps teams. Splunk brings together data from across the technology landscape and the associated tools — with full fidelity — at scale. The shared data and reporting make it easier for teams to determine and prioritize critical tasks, build shared KPIs, track progress and iterate as needed throughout the DevOps lifecycle.

2 **Adopt new security tools and processes that reduce friction for DevOps and security teams:** As organizations adopt DevSecOps, they may have to add on to their existing tools and processes — all of which need to fit within the existing workflows. Developers are not security experts — for them to continue delivering at a rapid pace, security tools and processes must integrate seamlessly into the existing DevOps toolchain, allowing them to work within their familiar integrated development environment. At the same time, the tools and processes must integrate with existing security workflows and SOC operations so that security teams can be true partners to their DevOps counterparts.

Splunk's approach: By bringing together data from any source, at any scale, from across the technology stack, Splunk provides contextual visibility to development, security and operations teams within their existing processes and workflows. In addition to providing discrete data views tuned to individual teams, Splunk allows users to build composite dashboards for a shared view across teams, providing leaders a comprehensive view of relevant metrics.

3 **Focus on automation:** Traditional application security approaches are usually heavyweight and gate driven, often requiring security professionals to perform them. These approaches don't scale for DevSecOps processes, which are agile and need continuous feedback. Just like DevOps, DevSecOps needs automation for speed and accuracy, helping to ensure that teams follow agreed upon protocols and best practices. When incidents do occur, automation is also essential for providing visibility and simplifying remediation. That said, automation should be done thoughtfully, with a focus on providing accurate, actionable results — not overloading systems unnecessarily, or flooding developers with false alerts.

For tasks that have to be done "out of band" and cannot be automated, teams need to create an iterative, predetermined schedule and put a system in place that ties results into the DevSecOps process.

Splunk's approach: Splunk provides automation in two ways. One is by seamlessly tying together data from across the various tools in the SDLC. This is simplified with prebuilt integrations for a wide variety of industry-leading tools, as well as Splunk's API-driven architecture, which provides connectivity to niche and company-specific tools. Ultimately, this automation helps reduce manual intervention and allows teams to offload repetitive tasks so they can focus on new and more innovative use cases.

The second is Splunk's support for larger automation initiatives around DevSecOps. By providing visibility into the health and functionality of these automations, Splunk helps mitigate the challenges created by black box processes, including predictive analytics that can call out issues before they happen.

4 **Ensure continuous, shared visibility:** Visibility and feedback need to be contextual and end-to-end — from when a feature is defined to when it is in production — and given as fast as the code that moves through the system. Both the development and operations teams need this visibility in their toolchain and existing processes, such as ticketing systems or Slack notifications. Security teams should also have visibility into all relevant metrics within their own processes and toolchains so that they can partner with their development and operations counterparts, and have access to all information necessary for tackling any security issues that arise post-production.

Splunk's approach: Because Splunk collects data across all tools and technology stacks, it can provide contextual visibility into the applications and the infrastructure they run on, as well as how individual stages of the process connect within the entire pipeline. In addition, Splunk extracts actionable insights with built-in AI/ML, streamlining the workflows for development, operations and security teams. Capabilities such as risk-based alerting prioritize incidents, reducing alert fatigue and helping ease security for developers.

5 Treat all security vulnerabilities as quality defects: Organizations often maintain two types of findings — security and quality — in two different locations. Not only does this practice reduce visibility, it often results in developers deprioritizing security defects. To address this issue, organizations need to have security and quality findings in one place, which provides an accurate shared view of security posture and helps the development team treat both quality and security issues with equal importance.

Splunk's approach: Because Splunk can pull data from across the DevOps and security toolchains and provide consolidated, shared dashboards, it allows teams to access a common repository and accurate view of security and quality defects, in real time. This shared view can help ensure that significant security defects are addressed early on to avoid costly rework at production time, and with the input of all involved teams.

6 Expand/strengthen post-incident response strategy: While security issues inevitably arise in production, having full contextual visibility from when a feature was defined will help teams quickly identify the problem. And because of the ephemeral nature of cloud architecture, it's also critical that they have full fidelity tracking on every single interaction. Even when an incident is assigned, security response and resolution teams might still need to collaborate — and having shared tools and visibility will only help drive better and faster resolution.

Splunk's approach: Beyond development, Splunk provides visibility to all incident data accompanied by built-in tools for strong incident response. That means SREs, who are typically on the front lines, have access to all the data they need to analyze security incidents. Splunk allows them to route alerts to the right people, assign response, and monitor case status and progress. When a case is assigned to a security specialist, Splunk ensures that all of the forensic investigation data already completed by the SRE is available to them, eliminating duplicate efforts. Splunk also provides end-to-end visibility from the time the features are defined, which allows security specialists to understand the incident without burdening the developer. Even when developers are not needed in the resolution process, they still have visibility into it, allowing them to understand the security impact of their code in production, and helping them define and prioritize security requirements for future projects.

Applying DevSecOps

There are potentially thousands of ways to use DevSecOps across numerous industries and verticals. However, almost all fall into one of three main categories: securing the development factory, supporting the creation of more secure apps, and securing apps in production.

1 **Secure the development factory:** For developers to be successful when working within the DevOps toolchain, their environment needs to be secure and resilient. But the DevOps toolchain involves a multitude of point tools supporting various discrete functions. That complexity is further compounded by a growing reliance on open-source software for building apps, and on adopting decoupled and ephemeral architectural patterns.

How Splunk helps: Splunk connects telemetry across these numerous tools and analyzes data patterns using AI/ML to create readable, risk-based alerts. This helps companies ensure that their employees are adhering to security policy no matter which tools they're using, while minimizing the noise from false alerts. Splunk also supports automated incident response to simplify remediation.

2 **Build more secure apps:** Building more secure apps means addressing security at every layer of the app — app components, cloud services and OSS libraries that apps rely on. Addressing security at every phase also includes the app's custom code, API interactions between different services, images built and deployed, and the infrastructure — increasingly in cloud/containers — on which the code runs.

How Splunk helps: Splunk can get logs from all of these layers in real time and help track the activity pipeline from feature definition to release, and on into production security incidents. This in-depth contextual visibility is provided through shared real-time dashboards and within existing tooling. As a result, developers can not only build more secure code, but also correct policy and security violations in real time during the process. This shared understanding across the teams also helps determine, track and measure optimal coding practices throughout the SDLC.

3 **Secure apps in production:** When a security incident does occur after deployment, the SRE and security team are typically responsible for remediation. But effectively remediating an issue is often complicated, not only due to the pace of development, but because developers need to understand both the earlier stages of the development cycle and the later stages of the in-use cycle.

How Splunk helps: Splunk tracks the entire activity pipeline, which accelerates both security investigation and incident resolution, and makes these processes more efficient, effectively reducing the churn between developers, SREs and security teams.

DevSecOps is fundamental to success in the Data Age

Security has become an even more critical concern to organizations as threats continue to evolve. At the same time, organizations face more pressure to either accelerate app production and development, or risk losing their competitive edge to speedier and more nimble players.

DevSecOps provides an answer to both challenges by ingraining security testing in the software development lifecycle without slowing the pace of DevOps. Because DevSecOps addresses security risks and vulnerabilities from the beginning, organizations can quickly mitigate or altogether avoid damaging and costly surprises that might have otherwise appeared in later phases. Organizations adopting DevSecOps can also implement continuous security, meaning their assets remain protected around the clock, every day of the year.

Splunk can help facilitate and scale these new DevSecOps approaches by providing end-to-end visibility, integrating the right data into relevant work streams and ensuring new tools and processes work well for both security and DevOps teams. By taking a holistic approach, Splunk ensures that the solutions work— not only with existing DevOps processes, but also with the security teams and their practices within the SOC.

Looking ahead, it's likely DevSecOps will not only occupy a bigger place in the development process, but also become more fundamental to the overall success of companies and their ability to thrive in the Data Age. While building and sustaining a DevSecOps culture won't happen overnight, rethinking the development pipeline with a DevSecOps focus will help organizations boost productivity, reduce risk and cultivate a superior security posture right from the start.

To learn how Splunk can help support a DevSecOps initiative, contact your Splunk Sales Specialist.

Learn More

turn data into doing®