tripwire

SOLUTION BRIEF

# Integrating Security Into DevOps Without Losing Momentum

> "The days are over when we just reviewed application security and the environment at the end of a project. Now we have to integrate that into daily work."
>
> — Gene Kim

**Continuous integration and delivery means DevOps teams bring products to market faster than ever. According to the Puppet/DORA *2017 State of DevOps Report*, the highest performing DevOps teams surveyed had a failure rate five times lower than their non-DevOps counterparts. And when there was a failure resulting in downtime, the teams were 96 times faster at recovering from that failure. It seems counter-intuitive, but moving fast and breaking things eventually leads to a lot less failure and downtime.**

But there's still a disconnect between DevOps and security workflows within most organizations. The rapid development of DevOps has left many security teams in a lurch, lacking the visibility they need. To make matters worse, DevOps teams tend to see security as an impediment to speed. And it can be, when security is treated as separate from the DevOps process—or even worse, as an inconvenient afterthought.

## Merging DevOps and Security Perspectives

The solution to this disconnect is the adoption of security controls that work within the DevOps lifecycle and integrate with the tools used in the continuous integration and continuous delivery (CI/CD) workflow. It's time to shift security assessment left and embed it into your DevOps process. If you have not yet managed to integrate full assessments of your application infrastructure into the DevOps CI/CD pipeline, that is where many organizations are now headed.

In a 2017 DevSecOps Community survey, more than half either somewhat or strongly agreed that "Security is an inhibitor to DevOps agility." So if teams can't have both security and DevOps agility, it seems they have to give one of them up. So, how do we resolve this conflict and bring DevOps and security together? Tripwire for DevOps is a SaaS that integrates security controls into the DevOps lifecycle, providing visibility into the security state of underlying application infrastructure.

## DevOps Teams Focus on Speed

DevOps emphasizes removing hand-offs in order to achieve the fastest possible development cycles to bring products to market faster than competitors. From the first line of code to deployment into production, a typical DevOps strategy would naturally prioritize having a single team handle all work to minimize delays. If DevOps teams need to hand their product off to a security team, they're left waiting for approvals. This bottleneck is made worse by the current cybersecurity skills gap, which leaves security teams with inadequate training and resources to complete those approvals quickly.

## Security Teams Can Get Left Behind

The main complaint of security teams working with DevOps teams is that they don't have enough visibility into the DevOps pipeline. They've typically got one or more tools they can use to assess their applications for vulnerabilities, but those tools don't extend into the DevOps toolchain so they can't see what DevOps teams are doing.

## The Answer is Visibility for Both Sides

DevSecOps is about shifting security processes to the left. Ideally, DevOps and security teams work together to address the concerns of external stakeholders. The work in their CI/CD lifecycles is automatically checked at every stage, so it goes through the pipeline already satisfying stakeholder requirements. Shifting left means remediation is quicker and less costly than having to go back and fix issues once in production.

## Do You Have a Security Blindspot?

There are a number of ways you might have security blind spots in your CI/CD process. One example is static file analysis. If you scan your containers for known vulnerabilities while they're not running, you run the risk of missing vulnerabilities that only become apparent when your containers are running. Oversights like this are what cause devastating breaches like the one Equifax suffered in 2017, wherein over 200,000 customers' credit card numbers were exposed.

That breach was caused by CVE-2017-9805, an exploit in Apache Struts. Struts is a framework used for creating Java web applications, which the Apache Software Foundation had patched via patch S2-045 for the vulnerability six months prior. This is one reason why containers must be tested while spun up through dynamic rather than static analysis.

## Dynamic vs. Static Analysis

Security breaches like the one Equifax made headlines with in 2017 are avoidable with Tripwire for DevOps. Breaches like Equifax's occur when teams only scan their containers for vulnerabilities in a static state. This is because some vulnerabilities only emerge when a container is running.

The best case scenario for those using static analysis is discovering and remediating these vulnerabilities during production. Tripwire for DevOps uses a sandbox environment to spin up containers in the cloud for dynamic analysis with all apps running so you can discover and remediate vulnerabilities *before* production. If you're finding known vulnerabilities in production, then you're finding them too late.

Static analysis is just one way your CI/CD process can go sideways because of a lack of security integration. DevOps teams can also find it hard to deploy adequate security because they don't have solutions that integrate with their DevOps toolkit or because they simply haven't managed to make the cultural shift toward a proactive involvement with their organization's security teams.

## Tripwire for DevOps

Founded by DevOps leader Gene Kim in 1997, Tripwire has set the industry standard for file integrity monitoring (FIM) and security configuration management (SCM) with Tripwire® Enterprise. As Tripwire's newest stand-alone SaaS solution, Tripwire for DevOps marks a significant shift in how we help organizations and agencies implement powerful cybersecurity: Think of it as Tripwire at the speed of DevOps.

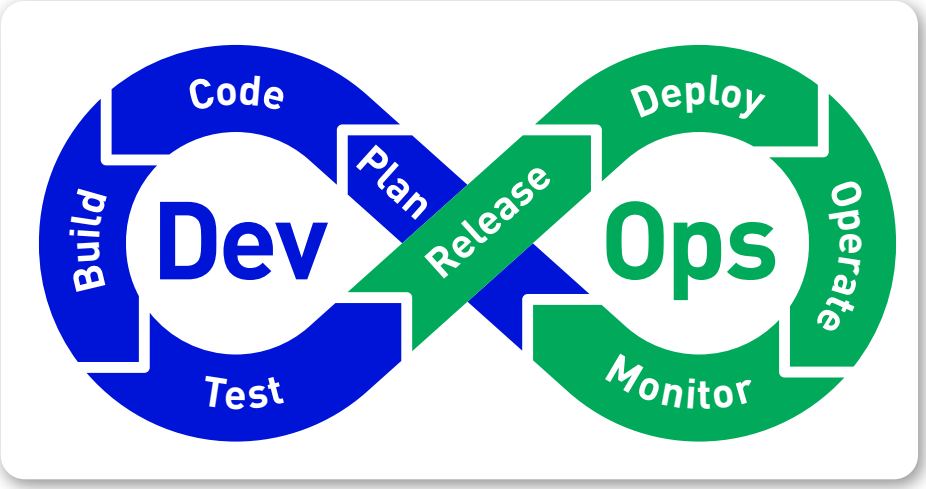


**Fig. 1** Effectively move security to the left with Tripwire for DevOps. .

Tripwire for DevOps is a comprehensive security SaaS solution that runs both static and dynamic analysis for vulnerabilities on container images in a sandboxed cloud environment. It equips DevOps teams with a complete security assessment of new application builds as they move through the CI/CD toolchain from development to production. This provides a quality gate that teams can use to pass or fail applications builds based on customizable security standards.

## Full DevOps Toolchain Integration

Tripwire for DevOps integrates into your existing toolchain and provides easy-to-understand results within the interfaces and scripts you already know. It's a fully self-contained SaaS that doesn't require you to purchase or access any other security products to operate.

Security shouldn't slow down continuous integration and delivery. Tripwire for DevOps provides native capabilities for system vulnerability assessment, as well as integrations with the popular CI/CD pipeline tools and REST API availability for custom integrations.

- **Docker**: Push your Docker images to Tripwire for DevOps' hosted Docker registry. Tripwire for DevOps can also periodically scan external image registries like Docker V2 and Amazon ECR.
- **Jenkins**: Tripwire for DevOps can act as a quality gate in Jenkins (as well as other build systems) to pass or fail new builds based on vulnerability scans. The results of the scans can be viewed inside the build servers' own native interfaces.

## Summary

It's time to take a smarter approach to DevOps security. Tripwire for DevOps makes it easy to reduce cycle time from code to deployment while conducting dynamic, comprehensive scans to catch and fix vulnerabilities before they make it into production. Learn more about Tripwire for DevOps by downloading the datasheet *"Tripwire for DevOps: All-in-One SaaS Security."*

### Ready for a Demo?

Let us take you through a demo of the Tripwire for DevOps and answer any questions you have. Visit **tripwire.com/contact/request-demo/**


SBISDO2b 2009