# INFORMATION SECURITY IN THE DEVOPS AGE:

## ALIGNING CONFLICTING IMPERATIVES

There's a saying in the IT world that if you consider a software release an event, then you're not doing DevOps. That's because DevOps involves teams of developers, testers and operational specialists working together with one mission in mind: rapid, iterative deployments that result in faster time to market, lower failure rates and shortened lead time between fixes.

What's not to love about that? Let's face it, DevOps — where software developers work jointly with operations — has taken the IT world by storm, evolving from what many thought was just another bleeding-edge experiment to an absolute requirement for businesses and government agencies of all sizes. DevOps is quickly becoming the default development methodology for government agencies. Forty-two percent of states are now adopting DevOps and another 37 percent of states have DevOps pilots underway, according to the latest survey of state CIOs from NASCIO.

**42%**
of states are now adopting DevOps

**37%**
of states have DevOps pilots underway

But business leaders and government executives shouldn't start celebrating just yet. The power trio of development, testing and operations is missing a beat — threatening the integrity of a much larger orchestration. While the focus on better code is certainly a major security benefit of DevOps, important questions remain about the role of information security and its integration into the DevOps cycle — and how best to improve visibility into potential security risks.

### OPPOSING PRINCIPLES

The benefits of DevOps are undeniable, but the current model may be creating security blind spots in applications that could have ramifications for the entire enterprise. This is not because of any fundamental flaw in the DevOps methodology; rather, it is a symptom of one of the longest-standing disconnects in IT history — the differing and often clashing cultural roots, vocabularies and behavior of developers and information security professionals.

DevOps teams are wired to move quickly, stay agile, keep sprinting. Information security specialists are trained to focus on control and stability. When information security leaders see change and churn, it breeds anxiety about system — and job — security.

Even after decades of experience and lessons learned, security still tends to be brought into the development cycle at the last stage — if at all. It may not be what used to be called "bolted on," but security considerations have certainly not "shifted left" far enough toward the development process. Ironically, treating security as a late-stage gate check is actually the antithesis of secure DevOps, and DevOps in general, and actually slows the development cycle.

## 5 TIPS FOR BRIDGING THE DEVOPS INFOSEC GAP

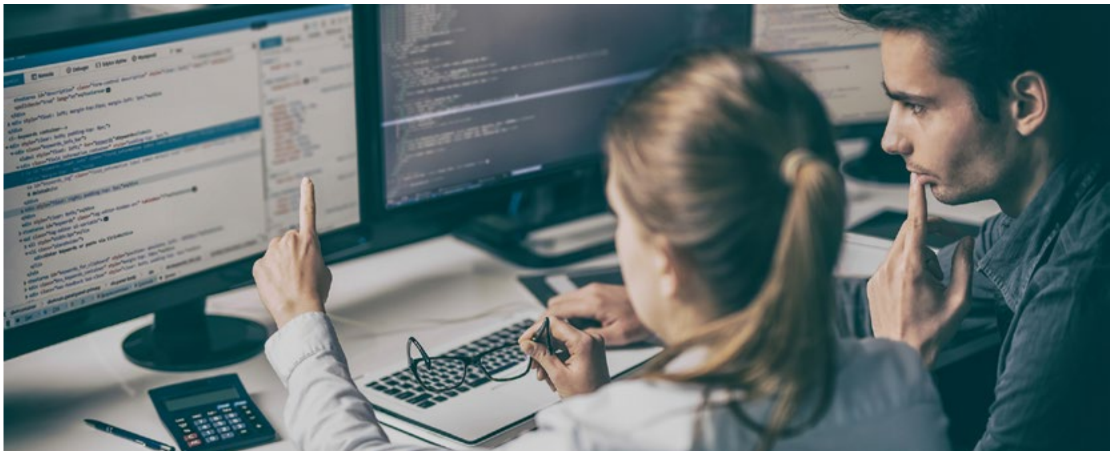
Security professionals must talk with developers about the security of their processes, and then define shared goals and collaboration opportunities. Here are some tips to achieve buy-in from your developer colleagues.

**1. Automate Security Testing:** Accelerate DevOps through automated security testing of container images during each new build from with CI/CD systems.

**2. Eliminate Code Defects:** Eliminate blind spots by conducting an inventory of container image "bill of materials" and detecting vulnerabilities and malware. Developers can then quickly remediate those defects before the app is pushed into production. For developers, security is about code quality.

**3. Reduce Operational Costs:** The cost to fix a software bug during the implementation stage is **roughly six times more expensive** than when it's identified in the during design, according the Systems Sciences Institute at IBM. And those costs increase exponentially as they are discovered later in the SDLC as software moves from design through maintenance, due to the increased complexity of implementing changes during production and identifying application owners to issuing corrective measures. Remediating vulnerabilities before deployment is critical to lowering overall security administration and labor costs.

A large part of the disconnect stems from vocabulary and orientation. Security professionals who talk to their DevOps brethren about security are often met with distracted, vacant stares. But talk to a DevOps devotee about improved code quality and fewer software defects, and they will engage with passion and enthusiasm. This is where security leaders have an opportunity to adapt, and become the force for change — and build the connective tissue that is currently missing between the two IT communities.



## SHIFTING LEFT

One place to start is for security leaders to engage DevOps teams in ways that change the flawed perception of security as primarily a compliance function that occurs only on occasion. The message should be that security is an integral part of DevOps success. By "shifting left" — that is, focusing on prevention rather than detection of vulnerabilities — security leaders can shift the conversation with DevOps teams to work together to reduce vulnerabilities from being embedded and shipped in the first place, which in turn, allows DevOps teams to work more efficiently.

One example of how to make this shift involves understanding the role of containers in application development. Containers speed application development and deployment by providing developers self-contained packages that have everything necessary to run an application: the application, dependencies, libraries, binaries and configuration files. This ensures that software runs reliably when moved from one computing environment to another across the software development lifecycle (SDLC).

The adoption of container technologies has exploded. In the case of the open source Docker—the leading container technology for the past three years — adoption has grown 40 percent year over year, according to recent Datadog user data. Docker offers more than 500,000 apps in its community registry and has recorded more than 8 billion container downloads since its inception.

While DevOps teams love containers, they can create blind spots and hidden risks for information security professionals. In Docker Hub alone, the average container image contains more than 40 vulnerabilities, according to research gathered by Tenable. Of organizations with containers currently in production, only 18 percent perform image scanning for vulnerabilities, according to recent survey by Anchore Inc. This represents a massive gap in security as developers increasingly download and use open source components to assemble their applications.

### 4. Eliminate Wasted Development Time:

Containers consist of multiple layers of software functionality — so ensure security tests provide layer intelligence to identify if and when security issues in lower layers are automatically mitigated in a higher layer to avoid frequent and costly false positives. Proper **automated testing** of container images early in the DevOps process will help eliminate the need for developers to work on fixes to non-vulnerabilities.

### 5. Develop & Publish Container Security Policies:

Notify developers immediately when a container image assessment exceeds an organizational risk threshold, and allow developers to take direct action from within their systems with specific security guidance.

## NEW APPROACH, NEW TOOLS

The short-lived nature of containers, lack of IP addressability and credentialed scans, and inability to remediate vulnerabilities all present special challenges that mean securing containers requires a different approach. Scheduling traditional vulnerability scans, for instance, requires multiple credentials used by privileged users. They can also tax network resources. And remediating vulnerabilities becomes more difficult without full visibility into containers.

One key way for security leaders to work with DevOps is to integrate vulnerability remediation into what are known as the Continuous Integration and Continuous Deployment (CI/CD) tool chain. CI is focused on building the application — the development process — and CD is focused on deployment after testing.

Fortunately, a new generation of automated tests can quickly identify coding vulnerabilities in ways that support the needs of DevOps and information security teams. With products such as Tenable.io Container Security, IT security teams are able to perform container assessments in less than 30 seconds during each new software build. Likewise, customizable dashboards and alerts for DevOps teams means developers get the information they need to remediate identified vulnerabilities early in the development process. This comprehensive view provides a detailed assessment of container image risk.

Moreover, IT security professionals are now able to protect containers from newly identified threats, using Tenable.io Container Security to monitor a wide range of external vulnerability databases. Container images are automatically re-tested as new vulnerabilities are identified to rapidly respond to emerging risks.

In addition, Tenable.io Container Security also ensures container images will be compliant with established cybersecurity policies before they are released into the registry. Now, IT security professionals can notify developers immediately with specific remediation advice when container images exceed organization risk thresholds.

The key for security leaders, of course, is to show DevOps teams that improving code quality and security can be done concurrently without erasing the time-to-market benefits of DevOps. In the language of DevOps, insecure code should be looked upon as just another software defect. Security leaders will gain greater alignment with their DevOps counterparts if they can establish common ground where shared intelligence, language and purpose improve speed to fulfillment and security.

*Learn more about how tools such as Tenable.io Container Security can improve your organization's agile development and security.*

THIS ARTICLE WAS PRODUCED BY cyberscoop | statescoop FOR, AND SPONSORED BY, tenable.