FORTINET®

Removing Security Barriers to DevOps

What DevOps Leaders Should Look for in a Solution

Table of Contents

Executive Overview: DevOps Increases Agility, but Security Requirements Slow Cycles

DevOps offers the promise of faster time to market and the automation of many previously manual processes. But it also presents security risks. One study finds that 92% of DevOps teams have seen at least one vulnerability slip into production in the past 12 months.[1] This is a serious problem.

To address the unique vulnerabilities of a DevOps environment, DevOps leaders often work with their cybersecurity teams to "retrofit" DevOps security tools onto an existing security architecture that may already have elements that are not integrated. In other cases, they use the built-in security tools provided by each public cloud, which puts each cloud into its own silo. This disaggregation creates fragmented visibility and control, communication gaps between the DevOps and security teams, and often, manual security processes.

While the security needs of DevOps are unique and must be addressed, end-to-end integration is the best solution for all network security—including for the DevOps infrastructure. Multi-cloud security, cloud workload security, and container security should be a part of a comprehensive security architecture that enables centralized visibility and control as well as automation of security processes. The solution should also acknowledge the need to confront an increasingly complex threat landscape, including east-west threats that propagate across workloads.

"How are we supposed to expect our employees to focus on security when the higher-ups are putting pressure on them to keep producing at a high volume? It needs to start at the top."[2]

An Integrated Security Architecture: Building on a Stable Foundation

Since DevOps cycles are driven by automation, any required manual work slows them down. Too often, manual security processes necessitated by disconnected systems can be a barrier to the success of DevOps projects. Manual processes also open the door to human error, including configuration mistakes.

To be effective, DevOps security must be built with the same agility that is essential to the DevOps methodology itself. Automation is key for at least two reasons. **First**, security processes must be automated to prevent unnecessary delays in DevOps cycles. **Second**, automation is necessary to respond to threats that move at machine speed. With that in mind, DevOps leaders should ensure that steps are taken to automate the following security processes:

- Integration of security into DevOps orchestration tools
- Security policy management across the infrastructure
- Automation of threat detection, prevention, and response

The only way to achieve true automation in these areas is to build a fully integrated security architecture. Here, DevOps leaders need a solution that delivers:

- **Single-pane-of-glass visibility** across and between DevOps environments as well as the broader security infrastructure
- **Centralized control of security policies** across the entire infrastructure, including DevOps environments
- **Extensive, aggregated threat intelligence**, including identification of new threats using artificial intelligence (AI) and machine learning (ML), shared across the integrated infrastructure

“Highly evolved [security] organizations are 24 times more likely to always automate security policy configurations compared to the least-evolved organizations.”[3]

Cloud Security Services Hub: Breaking Down Cloud Silos

A recent survey finds that 85% of companies operate in multiple clouds, and 39% have deployed DevOps processes and tool chains across their different cloud environments[4]—a number that continues to grow. Since different clouds have varying architectures and their management and security tools do not talk to each other, it becomes increasingly difficult to achieve an enterprise view of the entire DevOps environment. This increases risk for DevOps projects and for the organization as a whole.

Instead, today's complex DevOps cycles require a consistent and unified approach to cloud security for the disparate environments in which they operate. What is needed is a cloud security services hub, or transit network. The result is that security is split from application development to provide centralized, shared, and consistent security enforcement. In a distributed environment, a security hub securely connects different networks, locations, clouds, and data centers. Such an architecture brings a number of benefits to DevOps leaders:

- **Full visibility** of the entire DevSecOps environment for the DevOps team and the security operations center (SOC)
- **Consistent enforcement of security policies** across the environment—from multiple clouds to the data center
- **Secure connections** enforced between all locations, enabling DevOps processes to proceed uninhibited

> **"Bolt-on solutions are a thing of the past. Security is something you build, not something you do."[5]**

Cloud Workload Protection: Defending Against New Advanced Threats

From the perspective of threat actors, the vast amount of data stored in public clouds makes them a prime target. Additionally, the application programming interfaces (APIs) of each cloud provider offer bad actors a window into the underlying structure of applications being developed, helping them identify and exploit vulnerabilities.

As a result, DevOps leaders need to ensure that their applications are protected against two areas of risk:

- Traditional, internet-borne threats
- New threats that propagate across workloads or are introduced via the APIs and user interfaces (UIs) of cloud providers

92% of DevOps leaders report having at least one vulnerability slip into production in the past year.[6]

Configuration Management: Centralized Visibility and Policy Management

Another critical priority is to prevent the misconfiguration of systems—both on the cloud management platform and in the application components themselves.[7] To address misconfiguration and other security issues, DevOps leaders need a solution that includes:

- A reliable **virtual machine (VM)-based next-generation firewall (NGFW)** to manage threats coming from the public internet (north-south traffic)
- Centralized, standardized **security policy management for Infrastructure-as-a-Service (IaaS)** deployments at the workload level, the network level, and the API level to protect against east-west threats.
- Centralized visibility and policy management of **the configuration life cycle** and protection against unwanted, **unsupervised configurations** at the cloud-account level.

“Together, DevOps and cloud computing are a powerhouse. [T]hey are able to drive meaningful IT transformation that directly impacts business goals.”[8]

Intent-based Segmentation: Verifying Users and Entities

Once seen as a straightforward way to control access to critical assets, network segmentation is no longer that simple. With DevOps activities often spread across multiple clouds, DevOps leaders know these new complexities all too well. Users and entities no longer have static IP addresses, and ingress/egress ports are now less easily defined. Further, the notion of trust is constantly changing, and thus binary, "yes/no" models of trust are no longer adequate.

The answer is intent-based segmentation, where IT assets are segmented in line with business outcomes. Access to segments is implemented using controls like identity and access management (IAM). And even trusted users are inspected using methods such as user and entity behavior analytics (UEBA). In this case, intent-based segmentation helps organizations to answer three basic questions:

- **Where** are the segments demarcated? This is achieved by using logic defined by the needs of the business.
- **How** is trust established? This occurs using a model that is kept up to date using continuous, adaptive trust.
- **What** is used to enforce access control? This occurs using a consistent, holistic approach.

To provide the protection that is needed in a fast-moving DevOps environment, every network request must be inspected in real time, and trust must be continuously assessed for all users and entities. DevOps leaders need a solution that:

- Dynamically adjusts security policies based on logical roles
- Inspects both north-south and east-west traffic
- Effectively blocks lateral movement of attacks

“Today’s digital economy requires a security approach that allows data, applications, and workflows to move freely across a distributed network while avoiding an open environment where attackers can easily move and cause damage.”[9]

Container Security: Protecting the Entire Container Life Cycle

Container technology has provided DevOps teams with new ways to develop applications in a more modular and resilient manner. This is accomplished by separating different logical functions of applications into separate containers for reasons of code life-cycle management and scalability.

But containers are still considered an emerging technology space, and a lack of standardization means that different technology and standards are used to interconnect various services in container-based applications.

For this reason, many organizations discover that their current security tools do not support containers, and they are forced to purchase a point product for container security. This creates or exacerbates a disaggregated DevOps security architecture, adds manual work that can slow down DevOps cycles, and increases risk.

To address these security requirements, DevOps leaders need a comprehensive, integrated container security solution that is:

- **Container aware.** The NGFW should connect seamlessly with the container management layer and recognize the labels of different containers. The result: DevOps and security teams should be able to use container labels in setting policies.
- **Container enabled.** The web application firewall (WAF) should have the ability to create a native container image, integrating web application and API protection for DevOps applications.
- **Container integrated.** All of the protections of an integrated security architecture should be dynamically integrated into a container and inserted into the application chain.
- **Protective of container registries.** Container registries typically have few controls on new containers published to them, which leaves an opening to threat actors. As a result, DevOps leaders need to scan container images for zero-day threats and monitor for unauthorized additions to registries.

“[C]ontainers have become our blank puzzle pieces. We take it, make it what we need, and drop it in.”[10]

Conclusion: Achieving Scalable and Agile DevSecOps

While their performance is typically measured by metrics like time to market rather than the security of their applications,[11] DevOps leaders have a vested interest in a secure environment—namely, security incidents can cause major delays. But traditional security approaches simply do not work in the context of DevOps. Silos must be broken down, manual processes eliminated, and policies and protection standardized across the entire network.

Achieving the above requires a holistic approach that integrates every DevOps environment as well as the broader security infrastructure, automates security processes, and provides full, centralized visibility across multiple clouds and the corporate data center. Furthermore, security should be integrated into the base architecture of every DevOps project—rather than being added as an afterthought. And its capabilities should be broad enough to work with a wide array of cutting-edge technology such as application containers.

In short, like DevOps itself, DevOps security must be agile, scalable, and adaptable.

[1] "2019 State of DevOps Security Report," Fortinet, May 10, 2019.

[2] Daniel Newman, "5 Reasons DevOps And Security Need To Work Together," Forbes, September 30, 2018.

[3] "2018 State of DevOps Report," Puppet, accessed May 17, 2019.

[4] Steve Cowley, et al., "Assembling your cloud orchestra: A field guide to multi-cloud management," IBM, October 2018.

[5] David Linthicum, "Put security in DevOps first, not as an add-on," TechBeacon, accessed May 19, 2019.

[6] "2019 State of DevOps Security Report," Fortinet, May 10, 2019.

[7] Asher Benbenisty, "Don't Go Once More Unto the Breach: Fix Those Policy Configuration Mistakes," Dark Reading, October 30, 2018.

[8] Jaymin Vyas, "DevOps and Cloud: A Symbiotic Relationship," DevOps.com, November 7, 2018.

[9] Jonathan Nguyen-Duy, "Zero Trust is Not Enough: The Case for Intent-Based Segmentation," Network Computing, March 22, 2019.

[10] Don MacVittie, "You Have a Box: The Impact of Containers on DevOps," DevOps.com, November 1, 2018.

[11] "2019 State of DevOps Security Report," Fortinet, May 10, 2019.

FORTINET®

www.fortinet.com

399614-0-0-EN

January 1, 2020 10:16 PM

eb-removing-security-barriers-devops