CrowdStrike White Paper

UPLEVELING OF THE CLOUD INFRASTRUCTURE AND ITS IMPLICATIONS ON CLOUD AND CONTAINER SECURITY

*Parts of this document previously appeared as an **article** on Forbes.com.*

EXECUTIVE SUMMARY

Containers represent an upleveling of cloud infrastructure from physical and virtual machines toward applications. As a direct consequence of increased thrust on software and applications that enterprises across all verticals are placing as part of their digital transformation, container adoption has been on the rise with a large number of enterprises moving their production workloads from physical and virtual machines into containers and Kubernetes environments. The move toward containers is characterized by a change across two dimensions. First, security awareness, which is typically limited to the production infrastructure, is being extended to earlier stages of application development ("shift-left"). Second, the security signal captured at runtime, which is typically limited to infrastructure-level events, is being extended to include application-level events such as container context ("shift-up"). A modern cloud security platform must account for these fundamental changes and provide a holistic solution that effectively blends traditional security capabilities with ones required by modern environments.

A modern cloud security platform must provide a holistic solution that effectively blends traditional security capabilities with ones required by modern environments.

CLOUD AND CONTAINER LANDSCAPE

The shared responsibility model of the cloud requires the tenants to be responsible for their own assets (software applications that are core to their enterprise), while the cloud provider is responsible for managing the underlying infrastructure. The division is explicit and pronounced in public clouds, but it is also true in private cloud deployments where the central IT organization serves as the cloud provider for individual business units.

Various cloud delivery models built over the years — infrastructure as a service (IaaS), platform as a service (PaaS), container as a service (CaaS) and function as a service (FaaS) — are simply expressions of where that line of control is drawn along the software-hardware stack. With each iteration of the cloud, that line has moved up the stack from bare metal to virtual machines to containers and functions, with focus shifting away from infrastructure and moving toward the application. What constitutes infrastructure — the lower-level support layers underlying the application — has itself been undergoing change. In the case of FaaS, for example, the language runtime, embedded within the back-end execution unit, is considered infrastructure, compared to the higher-level application logic built on top of it.

The progressive upleveling of cloud infrastructure toward the application has been characterized by a change across two dimensions, as shown in Figure 1. First, the well-known shift-left movement extends security awareness to application development, redefining the historical roles and boundaries of development and operations teams. The second movement, which we call "shift-up," seeks to extend the runtime security signal to include application-level events crucial to addressing the emerging class of application-level breaches.

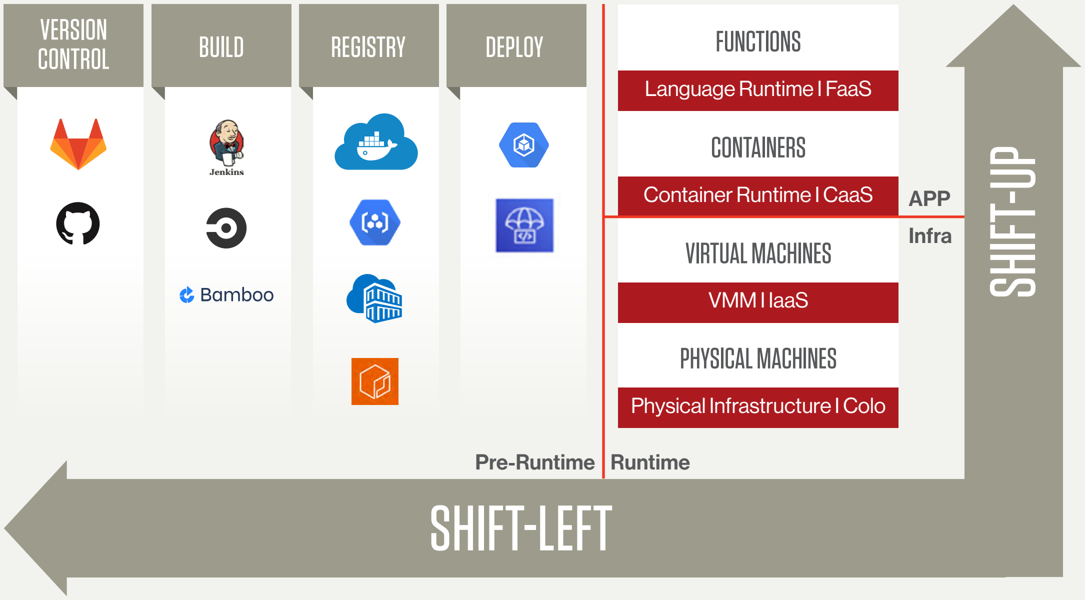

Figure 1. Emerging landscape of cloud and container environments where security awareness is being extended beyond the traditional production infrastructure toward the applications and processes surrounding them.

The rest of this white paper discusses each of these fundamental changes in turn and articulates how the next-generation capabilities of the CrowdStrike's Falcon Cloud Workload Protection platform helps organizations comprehensively secure their applications across various stages of the application lifecycle and across traditional and application-centric container environments.

SHIFT-LEFT: PRE-RUNTIME PROTECTION

Containers have created a new control point along the software-hardware stack. They effectively articulate a clear boundary between applications and the platform. What used to be a blob of software consisting of the operating system and applications tied together is now partitioned into the application and the underlying platform. By explicitly defining the composition of an application together with all pieces of data required to operate it, containers brought visibility and structure to the application lifecycle, from development through its operation and maintenance at runtime. Aspects of the application lifecycle that were previously invisible are being standardized through dozens of new patterns and the tools created in support of those patterns. Security can now be directly baked into these stages as extensions to that tooling before the application is deployed.

Because containers now include the specific platform dependencies required for the application, the developers responsible for building respective container images play a critical role in ensuring their validity and security. With developers now serving as administrators by default, they also continue to be responsible for patching and fixing issues in the workloads they are building and deploying. New versions of container images promoted into the registry are directly picked up by container orchestrators as part of their upgrade cycles.

IMAGE ASSESSMENT

Given the importance of capturing and addressing software defects earlier, it is imperative to have security baked into the software development lifecycle. An agile and well-operated application development and delivery pipeline would substantially improve the security of an organization by ensuring that vulnerable software is detected well before it is ever deployed.

Below is a brief outline of a few of the security checks that need to be performed as part of the application development stage to verify the suitability of containerized software for production use.

- **Binary Vulnerabilities:** Most applications have specific operating system and library dependencies. Since a containerized application links with its dependencies built into its container image rather than the ones on the host, any host-level monitoring and vulnerability scanning is clearly inadequate. It is also important to note container images need to be scanned periodically to verify against any newly discovered vulnerabilities. This is not unlike host-level vulnerability assessment, but given the requirement for high agility and typical lack of shells in containers, the process can be more involved.

- **Malware:** Several studies indicate a high incidence rate of malware embedded within container images, given their susceptibility to supply chain attacks. Some of that malware inevitably finds its way into internal containers. To ensure a high level of image hygiene, it is important to scan the images based on a well-curated vulnerability feed. Furthermore, simple hash-based verification is typically not effective against rapidly mutating malware versions, and machine learning-based algorithms need to be used.

- **Misconfigurations:** Containers articulate the overall structure and composition of the applications. In particular, they expose application context in the form of Dockerfiles and application manifests based on which application images are produced. It is necessary to extend verification to this rich metadata in addition to the application binaries and libraries.

- **Secrets:** Secrets — such as database passwords, tokens for cloud vendors or third-party API services, and internal authentication tokens for interservice communication — may be inadvertently packaged into container images. It is important to look for patterns representing secrets within container images while not causing false positives.

Secrets may be inadvertently packaged into container images, and it is important to look for patterns representing secrets within container images while not causing false positives.

SHIFT-UP: APPLICATION-LEVEL RUNTIME PROTECTION

Applications are the crown jewels of every organization, and protecting those assets is critical. Cloud providers aggressively invest in securing the low-level infrastructure side of the stack they manage. In particular, they tightly control the southbound interfaces consumed by the tenant software. However, when it comes to protecting the attack surface created by the northbound interfaces exposed by the applications, they do not assume any responsibility. Modern microservices workloads are especially vulnerable due to increased attack surface created by application decomposition. What used to be one monolithic application with obscure non-standard interfaces is now built as an often large composition of many microservices, each with its own set of web interfaces. That creates a gap where cloud tenants are underserved, and cloud security platforms are required to adapt and evolve in two important ways in order to address the gap — first, application-level security indicators need to be captured and analyzed, and second, system-level security indicators must be captured and analyzed even in environments that do not allow privileged access to the underlying infrastructure.

APP-LEVEL SECURITY ANALYTICS BASED ON APP-LEVEL INDICATORS

Having offloaded the infrastructure responsibility to the cloud provider, the tenant is primarily concerned with managing and securing the higher-level application layers of the stack. The nature of attacks is expanding to be more application-centric. The slightest vulnerabilities in the application layers are being exploited to manipulate the API behavior and cause service disruption, data leakage and account lockouts. Ultimately, applications are the units of concern for a CIO rather than infrastructure endpoints. This, in turn, requires the cloud security platforms serving those cloud tenants to closely monitor application-level events and indicators.

Fortunately, modern application structures based on containers and microservices provide the necessary control to detect and prevent such attacks. Containers, in particular, decompose applications into individual microservices, making visible what would otherwise be internal events deep within a monolithic application. In the microservice paradigm, enabled by containers, applications are developed as a group of cohesive microservices tied together through extremely well-defined web interfaces, often based on JSON objects that retain all relevant application context. They are usually built using the language and framework of choice by independent teams that are also responsible for operating them post-deployment. Each microservice typically invokes other upstream services in response to incoming requests. Key information about the application behavior, including its integrity, health and performance, can be gathered by tapping into these interservice interactions. A modern cloud and container security platform must capture, analyze and support detections and preventions based on the rich signal exposed by the modern microservice practices.

SYSTEM-LEVEL SECURITY ANALYTICS BASED ON UNPRIVILEGED SECURITY AGENTS

The cloud provider’s primary concern is to secure the infrastructure portion of the stack. While it is incentivized to serve its tenants, its primary concern lies in defending itself from potentially malicious software deployed by its customers. That makes for an interesting dynamic where the cloud providers need to enable their customers to construct secure application environments on top of the cloud infrastructure, yet they are unable to provide the necessary levers to access back-end hosts and infrastructure controls. At best, they may expose specific tightly controlled API points for visibility. In order to sufficiently protect applications deployed on such locked-down environments, a modern cloud security platform must be able to support system-level security without privileged access to cloud infrastructure hosts.

FALCON CLOUD WORKLOAD PROTECTION

The disruptive trends discussed in this document make security more important than ever. The CrowdStrike Falcon® platform is explicitly designed to account for these trends and make the adoption of the emerging container environments a seamless process. **Falcon Cloud Workload Protection** effectively increases the Falcon platform's breadth of support to encompass shift-left and shift-up trends while retaining the depth of its security capability.

Key capabilities of Falcon Cloud Workload Protection include:

- Falcon Image Assessment supports a wide range of verification techniques based on CrowdStrike's cloud machine-learning backend, including the ones discussed in this white paper.
- Deployed through a simple Kubernetes admission controller, Falcon Container provides comprehensive protection across all Kubernetes applications deployed to the target cluster.
- The Falcon admission controller transparently introduces the lightweight Falcon Container sensor into each application pod.
- The Falcon sensor runs alongside the application containers of each pod as an unprivileged container in order to support locked-down environments such as Fargate where privileged containers are disallowed.
- Key container-level information such as the Kubernetes namespace and the worker node where the pod is scheduled are made available through the Falcon UI for analysis.
- Falcon Container retains the same capabilities of the host-based Falcon sensor such as CrowdStrike Threat Graph®, detections and preventions.

What is occurring in the industry is an unlikely combination of traditional security focused on infrastructure endpoints with the addition of applications and their development processes. While these shifts pose new challenges to enterprises, they also lend unprecedented opportunity to optimize both efficiency and security of existing environments and processes. To learn more about how CrowdStrike can help you turn challenges presented by containers into an opportunity for competitive advantage, and enable DevOps teams to deploy applications with greater speed and efficiency, visit **crowdstrike.com**.

The CrowdStrike Falcon® platform is explicitly designed to account for disruptive trends and make the adoption of the emerging container environments a seamless process.

ABOUT CROWDSTRIKE

CrowdStrike, a global cybersecurity leader, is redefining security for the cloud era with an endpoint protection platform built from the ground up to stop breaches. The CrowdStrike Falcon® platform's single lightweight-agent architecture leverages cloud-scale artificial intelligence (AI) and offers real-time protection and visibility across the enterprise, preventing attacks on endpoints on or off the network. Powered by the proprietary CrowdStrike Threat Graph®, CrowdStrike Falcon correlates over 5 trillion endpoint-related events per week in real time from across the globe, fueling one of the world's most advanced data platforms for security.

With CrowdStrike, customers benefit from better protection, better performance and immediate time-to-value delivered by the cloud-native Falcon platform.

There's only one thing to remember about CrowdStrike: **We stop breaches.**

Start Free Trial

Learn more at **www.crowdstrike.com**