FORTINET

SOLUTION BRIEF

Fortinet Container Security

Executive Summary

Container technology enables developers with new ways to design and 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. Further, by breaking applications out into different logical functions—namely, microservices—container technology enables enhanced portability of microservices and applications across different public and private cloud environments.

Due to the radical shift in application architectures, many organizations find that current security solutions are inadequate to address the risks associated with container technologies. This gap forces them to purchase point security products to protect their container environments and force-fit them into the new application life cycle associated with container technology. The Fortinet container security strategy offers multiple solutions that address the entire life cycle of container-based applications, providing comprehensive protection against threats associated with the different threat vectors to which container-based applications are exposed.

The Key Attributes of Container Security:

- Container-aware security
- Container-enabled security
- Container-integrated security
- Container registry security

Teams need to be able to rapidly develop modular applications in containers—on-premises or in the cloud. Security should be as portable and consistent as the applications themselves.

The Benefits of Container Technology

Organizations use containers to break applications into even smaller pieces than virtual machines—pieces that basically provide autonomous functionality. They want to maintain each piece separately (including different versions and bug fixes) and scale each piece separately, as different pieces may require different levels of performance. These pieces are typically referred to as services—or increasingly, microservices.

In server virtualization environments, in order to manage repositories of images, virtual machines (VMs) have a set of metadata attributes that are visible at the hypervisor or virtual infrastructure levels—commonly referred to as “tags.” Similarly, containers also have associated metadata attributes that are typically called “labels.”

Currently, the most common container-format standard is Docker, which has both an open-source and a commercial implementation. To build an application using container technologies, an organization requires multiple, interdependent services interacting with each other—typically called a POD. Building applications requires multiple containers, and while they may or may not be grouped into PODs, they all must be interconnected for the application to properly operate.

Containers are connected by a service/application composition, which is normally performed as part of the orchestration process of bringing up a containerized application. This orchestration process dynamically assigns addresses for the different services and offers a service/name resolution capability for different services used to resolve each other. This is where Kubernetes enters the picture.

Kubernetes has emerged to become the most common container orchestration system, with the ability and instrumentation necessary to describe application compositions, service dependencies, service-scale requirements, service-availability requirements, and more. Kubernetes also provides the tools needed to independently manage the life cycle, scalability, availability, and performance of different services without interrupting the availability of an entire application.

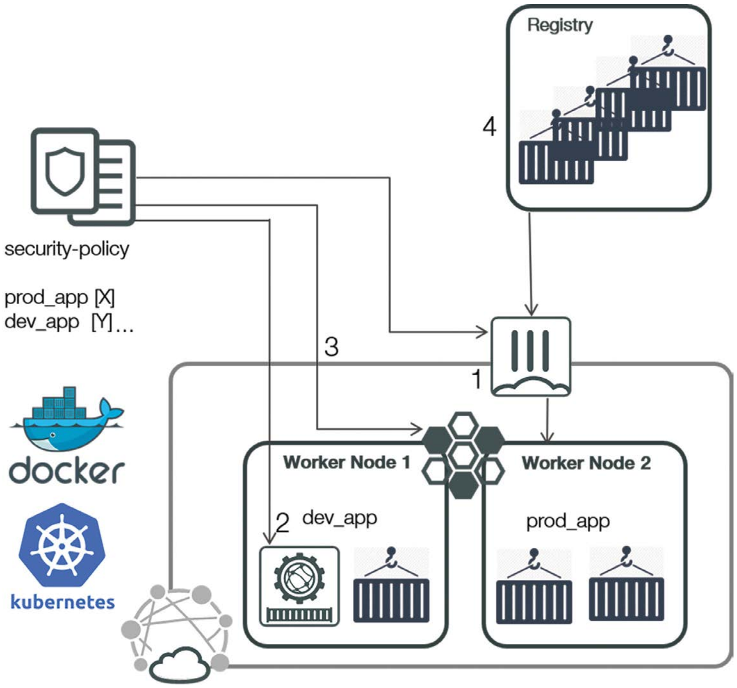

Figure 1: Protecting application containers throughout the application life cycle.

Security for the Entire Application Container Life Cycle with Fortinet

As containers are still considered an emerging technology space, and as different technologies and standards are being used to interconnect different services in container-based applications, there is a need for a comprehensive security solution to address the key attributes of container security.

The Fortinet container security solution solves these requirements in the following ways:

1. **Container-aware security.** The FortiGate next-generation firewall (NGFW) effectively connects to the container management layer and learns the labels of different containers. Security policies are label-aware and can use these labels to describe objects in security policies. This solution is primarily relevant for securing traffic in and out of the container infrastructure—namely, north-south security. FortiGate NGFWs offer Fabric Connectors that interface with major container orchestration systems to leverage metadata as security policy objects, including native Kubernetes, AWS EKS, GCP GKE, Azure AKS, and OCI OKE. When traffic leaves the boundaries of a containerized environment, it crosses a FortiGate NGFW that enforces the policy based on the container role. FortiGate also scans ingress and egress container traffic for vulnerabilities and file-based threats using an intrusion prevention system and advanced malware protection via FortiSandbox integrations.
2. **Container-enabled security.** Organizations can also leverage the FortiWeb web application firewall as a container image that can be bundled within an application chain. Since it is very typical to create microservices for web service-based applications, the ability to couple web application and API protection with microservice-based applications offers significant benefits to organizations building these applications. Developers can roll out security controls alongside their application development life cycle and port application security along with the other application services throughout the application life cycle to different environments. FortiWeb is currently offered as a native Docker

High-performing DevOps organizations report 46x more frequent code deployment, 96x faster mean time to recover, 440x faster lead time from commit to deploy, and 5x lower change failure rates than low-performing DevOps organizations.[1]

container as well as an AWS EKS marketplace offering. FortiWeb also integrates with FortiSandbox, adding an additonal layer of zero-day threat protection for incoming and outgoing traffic.

3. **Container-integrated security.** Much of the container-based internal application traffic occurs within the container host and is not visible in the network infrastructure. To ensure that each traffic flow is inspected, traffic flow between services must be modified within the application or a mechanism of service insertion. Security services are attached to the application composition to ensure that all traffic flow is visible to the security-processing services. These services can be either container-based or network-based (residing outside of the container infrastructure). The techniques and technologies used for inserting services alongside container-based applications are still emerging, and the ability to predict which technology will dominate is very limited. In this case, Fortinet partners with leading third-party providers of security solutions for container infrastructures, offering security solutions that integrate all attributes of container security into a single solution.

4. **Container registry security.** Container images are typically stored in public repositories known as registries, and there are few restrictions on the publication of new container images to them. This often leads to container images that are intentionally or mistakenly seeded with malicious code that is easily "pulled" from the registry by application developers. This situation introduces unnecessary risk to the application development process. FortiSandbox offers the application programming interfaces (APIs) and integration capabilities to specifically address the needs of container-based application developers. This helps them mitigate the potential risks introduced through their agile development methodologies.

Top-tier DevOps security organizations are 187% more likely to do security audit tracking, 96% more likely to do dependencies analysis, 63% more likely to scan cloud instances for misconfigurations, and 45% more likely to monitor and manage code commits than bottom-tier DevOps security organizations.[2]

Enabling a Secure, Holistic Container Strategy

Container technologies are rapidly becoming more popular as an application infrastructure and development technology. However, the risks they introduce are inadequately addressed by using traditional security tools. The ability to source a comprehensive container security solution that is compatible with a broad range of container orchestration systems is essential for organizations deploying any application on any container infrastructure in both public or private cloud environments.

Container security solutions from Fortinet fully address the expanded attack surface, enabling security to be integrated in the container application life cycle and allowing organizations to deliver more secure applications.

[1] "State of DevOps: Market Segmentation Report," Puppet, May 2018.

[2] "2019 State of DevOps Security Report," Fortinet, May 10, 2019.