Solution Showcase

Runtime Application Container Security with NeuVector

Date: July 2018 **Author:** Doug Cahill, Senior Analyst

Abstract: The broad adoption of application containers for both new applications and those being refactored into a microservices architecture has created the need for purposeful container security solutions. And as application containers move along the build-ship-run continuum into production environments, such solutions need to protect containers from being compromised from multiple vectors. Integration into the orchestration platforms that automate container management lifecycle is an immutable aspect of application container security that represents an opportunity to assure that containers are secured from registry to production. NeuVector provides an application container security solution focused on protecting the runtime integrity of containers from multiple attack vectors by automating the introduction of security controls via integration with container orchestration platforms.

Overview

Application Containers Are Moving into Production

The rise of application containers over the past several years reminds us of the rapid adoption of virtual machines (VMs) in the mid-2000s. The emergence of application containers and the automation platforms that manage them, including Docker, Kubernetes, and OpenShift, represents a new application development and delivery paradigm, one optimized for the IT agility businesses demand. But to what degree are application containers now moving out of the lab and into production environments?

Research conducted by ESG reveals that over half of respondent organizations are now running application containers in a production capacity, with 13% having an extensive number of application containers in production (see Figure 1).[1] Another 24% of the respondents shared that their organizations are not far behind, as they are already testing and planning to deploy containers to production in the next 12 months.

> …over half of respondent organizations are now running application containers in a production capacity…

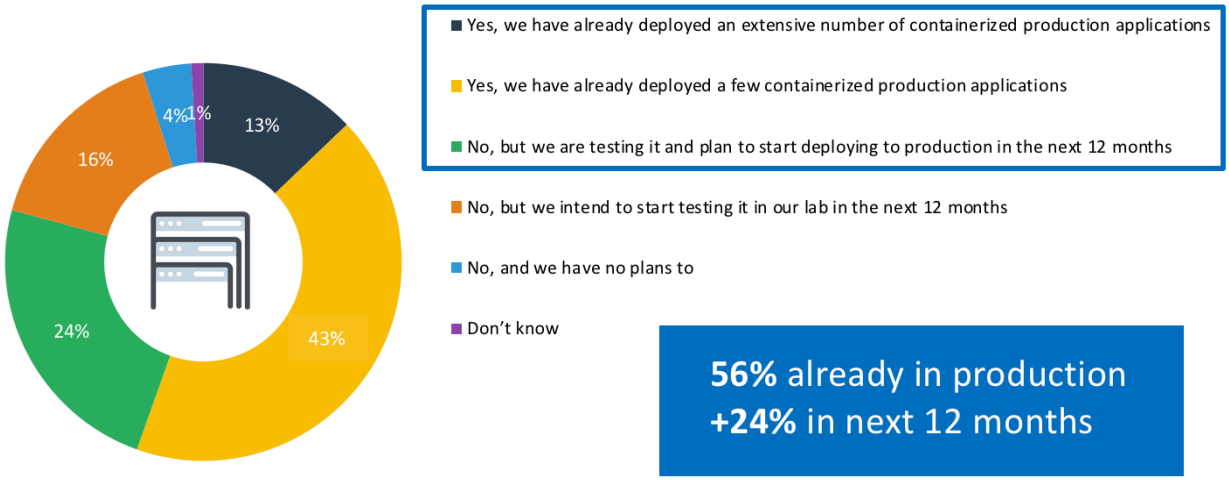
While initially containers were largely used by cloud-native organizations, ESG's research shows that containers are being adopted across industries and organizational sizes, with both new and legacy applications being containerized. In fact, 73% of organizations indicated that they are using or will use containers for new applications and some pre-existing "legacy" applications. These containerized applications are being deployed across the platforms of hybrid clouds, with 51% noting

[1] Source: ESG Master Survey Results, *Trends in Hybrid Cloud Security*, March 2018. All ESG research references and charts in this solution showcase have been taken from this set of master survey results unless otherwise noted.

that their container-based applications will be deployed on-premises, 21% in a public cloud platform, and 27% in a combination of public cloud platforms and private clouds.

Figure 1. Use of Application Containers in a Production Capacity

Does your organization use application containerization in a production capacity today? (Percent of respondents, N=450)



Source: Enterprise Strategy Group

Multiple Attack Vectors Put Containers at Risk

The rapid adoption of containers creates a need to understand ways in which containers and their application code can be attacked. Attack vectors employed by adversaries span the application container stack and include:

- The host as an attack vector represents a one-to-many entry point for threats in that all guest containers are susceptible to comprise when the underlying host has been attacked. Host attacks can originate from phishing attacks, container breakouts, or network attacks as described below.
- Container guests can be exploited (breakouts) when vulnerable containers make their way to production, including those with open source software components. Containers that have not been properly segmented are susceptible to unauthorized access from other containers and even external entities via port scanners seeking a soft target.
- The network can also be employed as an attack vector by attacks that move laterally and those that employ application-level methods such as vector SQL injections, cross-site scripting (XSS), denial-of-service, and zero-day attacks.

Shifting Application Container Security Right

Much has been said and written about shifting security left in the continuous integration and continuous delivery (CI/CD) pipeline, and for good reason; incorporating security practices in development and test phases and environments helps assure secure code and hardened configurations are delivered to runtime production environments. Because containers are quickly moving into production, container

Because containers are quickly moving into production, container security needs to follow suit and *shift right* with the use of runtime controls to protect containers from the multiple attack vectors that put them at risk.

security needs to follow suit and shift right with the use of runtime controls to protect containers from the multiple attack vectors that put them at risk.

Enable DevOps and SecOps Use Cases

The shift-left metaphor is typically associated with dev and test environments and the build phase of the container lifecycle. Use cases here include source code composition analysis, code analysis, vulnerability management, and configuration hardening, which collectively serve to reduce attack surface. As we shift container security right, we need to include SecOps use cases, including the segmentation of containers with container firewalls that understand the layer 7 application protocols containers use to speak to each other, as well as intra-container visibility into system activity. Container activity should be monitored for anomalous activity with associated alerts propagated for review by SecOps.

Secure the Automation Infrastructure

These DevOps and SecOps use cases are enabled and introduced via integration with the automation platforms that orchestrate the build-ship-run stages of containers, which is most often, but not exclusively Kubernetes. Because Kubernetes has a central role in performing tasks such as standing up container pods and replicating containers to auto-scale, the Kubernetes infrastructure, including the API server and master and worker nodes, represents a point of vulnerability. As such, the Kubernetes infrastructure itself also needs to be secured as part of a holistic approach to application container security.

As such, the Kubernetes infrastructure itself also needs to be secured as part of a holistic approach to application container security.

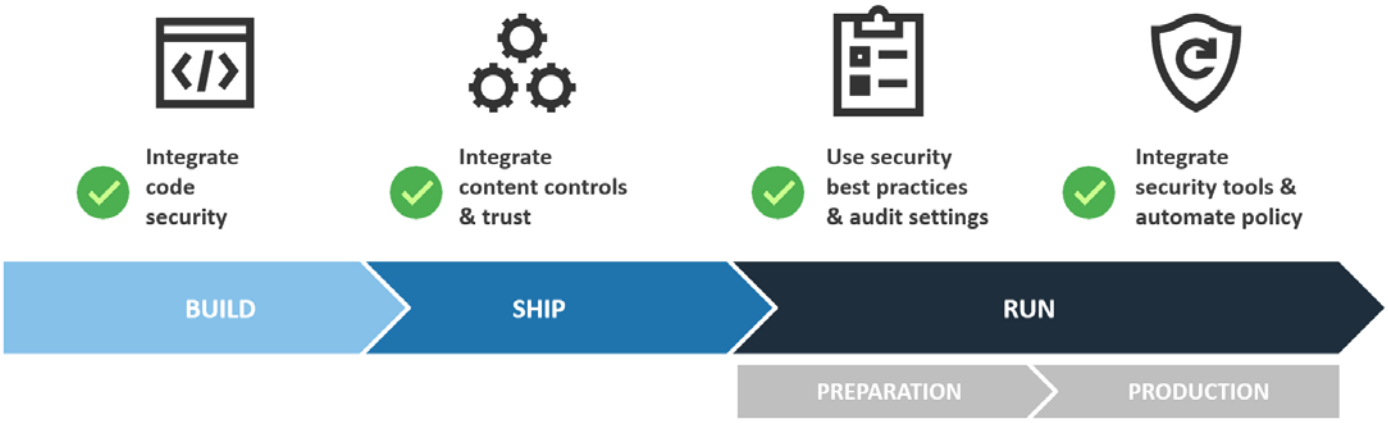
Secure the Rise in East-West Traffic Via Segmentation and Inspection

Microservices architectures that result in multiple discrete entities—containers—communicating via application protocols appreciably increase the east-west traffic between containers on the same pod, those on other pods, as well as with non-container services. Because of this level of traffic and the layer 7 application protocols in use, purposeful container firewalls are required both to segment container traffic per a trust-based security model and to inspect that traffic to detect and prevent application-level attacks and the lateral movement of threats.

NeuVector's Holistic Multi-vector Approach to Application Container Security

NeuVector offers a holistic approach to securing the container lifecycle with pre-deployment and runtime controls as well as the Kubernetes environment that orchestrates container management (see Figure 2). The product also dovetails with the continuous integration and continuous delivery pipeline to streamline the introduction of security and compliance checks and controls. The set of pre-deployment and runtime controls discussed below are designed to mitigate the risks associated with the attack vectors noted above.

Figure 2. NeuVector's Continuous Application Container Security Methodology



Source: Enterprise Strategy Group

Assuring Hardened and Compliant Containers Pre-Deployment

NeuVector provides the capabilities required to assure that application containers shipped to production have been vetted and meet regulatory compliance requirements. To do so, NeuVector reaches into container image registries to secure the build phase with the following measures:

- Software vulnerability scanning: Image scanning and analysis is performed to identify and remediate software vulnerabilities during the build process as well as in registries.
- Configuration assessments: Based on the CIS Docker benchmark, NeuVector assesses the configuration of registry-resident images to assure no unnecessary libraries are included and the image is one of a secure configuration.

Purposeful Application Container Firewall Implementation

NeuVector offers a purposeful firewall with a microservices design center which does not require agents or changes to application containers. Network connections between containers are automatically discovered and modelled, allowing customers to either enforce the current set of connections or create new rules that govern both inter-container and non-container communications. Container network segmentation is augmented by the ability to inspect network traffic with visibility into layer 7 (application protocol) traffic.

Deep network inspection into application protocols is employed to detect unauthorized and potentially malicious connections that could be indicative of an attempted application-level compromise or the lateral movement of threats along the kill chain. By being layer 7-aware, NeuVector is able to thwart attacks such as denial-of-services attacks that post HTTP (HyperText Transfer Protocol) get and post requests and brute force SSH (Secure Shell) attacks that attempt to gain access to systems.

Deep network inspection into application protocols is employed to detect unauthorized and potentially malicious connections that could be indicative of an attempted application-level compromise or the lateral movement of threats along the kill chain.

In addition to implementing RBAC (role-based-access-controls) for access to Kubernetes API server, NeuVector's application container firewall provides another means of protecting the API server from unauthorized access.

Maintaining System Integrity with Continuous Runtime Controls

Because containers operate, by definition, in a highly dynamic environment, a continuous approach is required to protect them against attacks launched via host, guest, and network attack vectors. With vulnerabilities having been eliminated prior to deployment to production, NeuVector extends vulnerability scanning into runtime by continuously scanning containers for vulnerabilities.

In addition to vulnerability scanning and the inspection of production container network traffic, core to NeuVector's runtime security functionality is the ability to lock down application containers by enforcing normal behavior based on an observed baseline. NeuVector establishes the normal runtime behavior of application controls based on system activity, including processes and file system access. Deviations from normalized baselines serve as the grounds for alerting on potentially malicious activity.

Integration with the Application Container Ecosystem

NeuVector integrates with the range of tools employed for modern application development, delivery, and management, including:

- CI/CD tools including Kubernetes to automate both pre-deployment checks and the inclusion of runtime controls.
- Notification and event management via support for SYSLOG and webhooks.
- Identity and access management (IAM) via LDAP integration for roles and SSO (single sign-on) via SAML (Security Assertion Markup Language).

The Bigger Truth

Application containers are front and center in the development and deployment of dynamic and elastic microservices-based applications. But containers represent more than a delivery vehicle for application code and another form of virtualization; the automation platforms that manage their lifecycle through the stages of build, ship, and run represent both an opportunity to automate security and a point of entry for threats. As such, a holistic approach to securing containerized application is required, one that protects both the Kubernetes infrastructure and fleets of container clusters from the attack vectors that put them at risk.

To secure the build-ship-run container continuum, security needs to shift left, and then, to assure vetted and trusted containers delivered to production are protected during runtime, container security also needs to shift right. Automation requires a continuous approach to these stages and a solution that integrates into the DevOps continuous integration and continuous development (CI/CD) tool chain employed by today's application development and infrastructure management teams. Given the rapid deployment of containers to production, organizations should not declare victory by deploying hardened containers. NeuVector offers such a purposeful solution, focused on securing the build, ship, and run phases of containers with an emphasis on securing both inter- and intra-container activity to protect containers from multiple attack vectors.