STRATA™
BY PALO ALTO NETWORKS

Protect Kubernetes Environments with CN-Series Firewalls

New breed of container firewall secures cloud-native applications

paloalto® NETWORKS

Table of Contents

Who's Driving the Container Revolution?

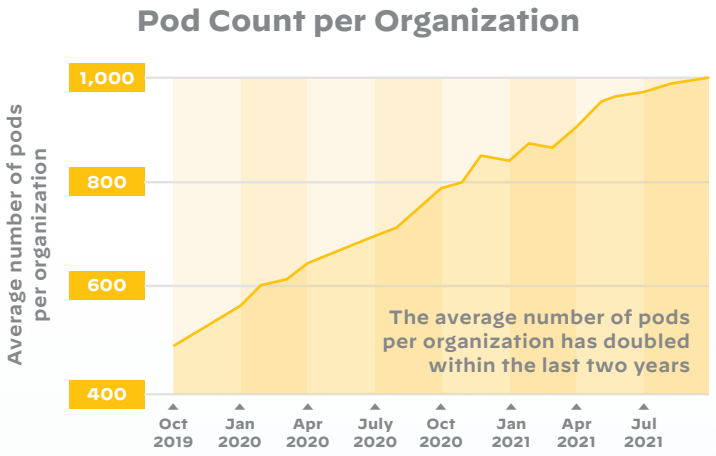
In just a few years, containers have become the predominant methodology for developing and releasing software applications in cloud environments. By 2023, more than 70% of global organizations will be running three or more containerized applications in production, up from less than 20% in 2019.[1]

That explosive growth rate is mirrored by the rapid adoption of Kubernetes, which has emerged as the de facto standard for container orchestration.[2]

Although developers were once the primary proponents for containerization and Kuberetes, IT operations is now getting into the act. In many organizations, IT Ops is assuming responsibility for containers from platform architects, developers, DevOps teams, and others. However, security teams are still under pressure to secure their applications before releasing them into production.

Did You Know?

"Within the past two years, the average number of pods per organization has doubled, with a similar relative increase in the average number of Kubernetes hosts."[4]



Source: DataDog [3]

[1] Gartner research quoted by Janakiram MSV, "5 Modern Infrastructure Trends To Watch Out for in 2019," Forbes, Dec 20, 2018.
[2] "6 Best Practices for Creating a Container Platform Strategy," Gartner, April 23, 2020
[3] "2021 Container Usage Report," DataDog, October 2021.
[4] Ibid.

Containers pose unique challenges in the area of application security. Learn more in the following section.

The DevOps Security Dilemma

One reason for adopting DevOps practices is the ability to speed time to market through continuous integration/continuous development (CI/CD).

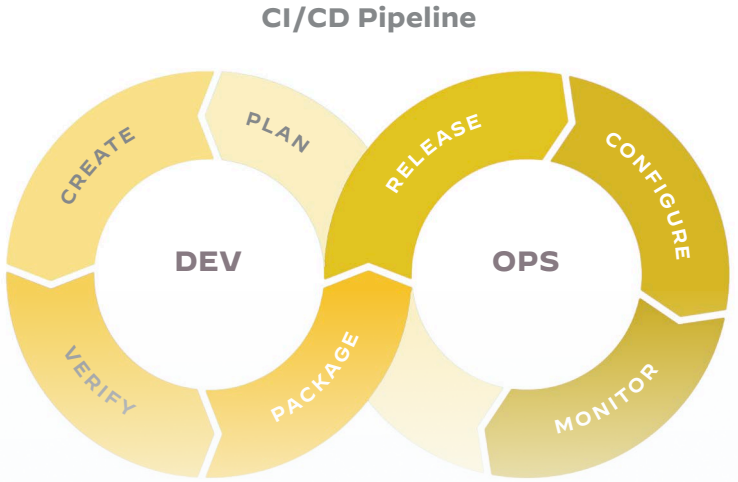
CI/CD pipelines include components such as code and image repositories, containers, build servers, and third-party tools, all working together to provide efficient integration and deployment. However, these complex dependencies and configurations contain potential vulnerabilities that allow attackers to exfiltrate data, disrupt production, and even bring down the entire infrastructure.

Because of their short lifespan and isolated nature, containers may seem like a secure option for running applications. But containers do not live in a vacuum—they depend on other non-containerized applications and therefore are almost never completely isolated. In fact, containers have their own security challenges due to new vulnerability points such as Kubernetes APIs and Kubernetes Orchestrator.

For these reasons, CI/CD pipeline security cannot be an afterthought—it must be designed into the application lifecycle.[5]

Did You Know?

Release cycles are speeding up, driven by the rise of DevOps, CI/CD tools, and agile development methodologies. Organizations with daily release cycles nearly doubled, from 15% in 2018 to 27% in 2019, and weekly release cycles have increased 20% to 28%.[6]



[5] "The Greatest Security Risks Lurking in Your CI/CD Pipeline," The New Stack, Jun 5, 2019.
[6] "CNCF Survey 2019," The Cloud Native Computing Foundation, 2019.

Understanding the risks associated with containers is key to developing effective security, as the next section explains.

Top Three Risks of Container Applications

1. Containers are subject to the same network-based attacks that plague legacy workloads

Containers are an innovative way to deploy applications, but they do not fundamentally alter the threat landscape from the application's point of view. Whether hosted on bare-metal servers, virtual machines, or containers, applications run on the same network stack and protocols and therefore face the same threats, for example, ransomware, cryptojacking, and botnets.

2. Containers lack protection against unpatched and unknown vulnerabilities

Application/host vulnerabilities are not always known. In some cases, they are discovered after years of existence. Additionally, when a vulnerability is identified and a patch is made available, it can take weeks or even months to patch hundreds of applications spread across the deployment. While agent-based security products help to identify and patch *known* vulnerabilities at the time of deployment, applications are helpless against *unknown and unpatched* vulnerabilities.

3. Fragmented responsibility compromises security.

Often, network security teams are not equipped with the right tools and expertise to secure containers without impacting CI/CD speed and agility. As a result, DevOps teams are often expected to secure the container infrastructure while network security teams do the rest. This fragmented approach to security creates gaps in the overall security posture, which attackers can exploit to laterally propagate threats in the environment, escalating the rapid spread of infections.

Did You Know?

Log4j is a ubiquitous piece of software used to record activities in a wide range of systems found in consumer-facing products and services. Recently, a serious vulnerability in Log4j was disclosed, posing a severe risk to millions of consumer products to enterprise software and web applications. This vulnerability is being widely exploited by a growing set of attackers.[7]

[7] "FTC warns companies to remediate Log4j security vulnerability," Federal Trade Commission, January 22, 2022.

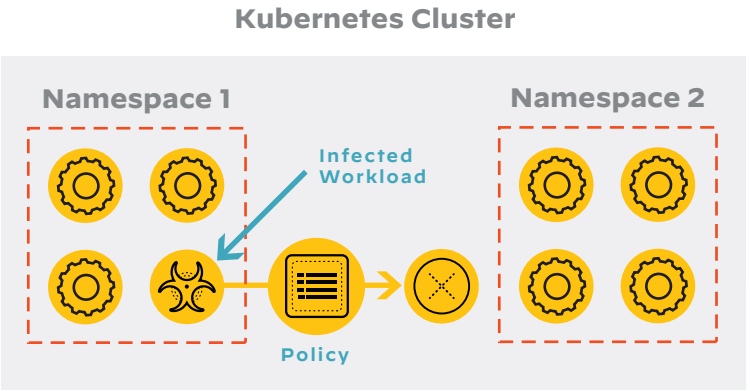
To secure Kubernetes clusters, you cannot be on the outside looking in. See what we mean next.

Namespaces: Powerful Tool for Cloud-Native Security

While Kubernetes creates challenges for traditional security tools, it also presents opportunities to enhance security by taking advantage of native constructs, most notably, namespaces.

Kubernetes namespaces help to simplify cluster management by making it easier to apply certain policies to some parts of the cluster without affecting others. However, they are also a valuable security tool. Security teams use namespaces to isolate workloads. This approach reduces the risk of attacks spreading within a cluster and establishes resource quotas to mitigate the damage that can be caused by a successful cluster breach.[8]

Forward-looking security architects want the ability to secure traffic that crosses namespace boundaries or travels outbound to legacy workloads. However, doing so requires knowing the internal state of objects such as namespaces, pods, and containers. Because that information is not available outside the environment, the only effective solution is to take the security solution inside the Kubernetes walls.



Security policies based on namespaces prevent spread of exploits within a physical cluster.

Did You Know?

Namespaces are virtual clusters running within a physical Kubernetes cluster.

[8] "Kubernetes Security Best Practices," Twistlock, June 6, 2019.

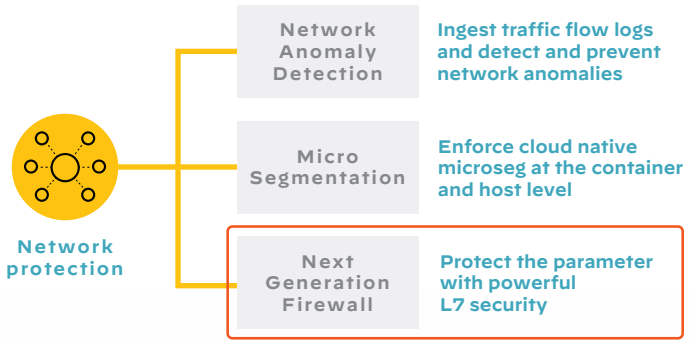
Kubernetes network policies are a powerful tool for container security, but they have limits. Read on.

What Kubernetes Network Policies Can and Cannot Do

Tried-and-true techniques such as policy groups, access control lists, and port blocking are important parts of security for both on-premises and cloud deployments. Kubernetes network policies allow developers to specify how a namespace is allowed to communicate with other network entities such as other namespaces, endpoints, and network services.

While Kubernetes network policies are used as high-level access control lists to allow or deny traffic based on Layer 3 and Layer 4 header information, most network-based attacks exploit legitimate permitted network connections to move laterally within the environment. Additionally, Kubernetes network policies alone cannot protect against unknown vulnerabilities such as the Apache Log 4j.

That is where NGFWs come in. NGFWs provide comprehensive Layer 7 runtime security through application-level inspection, intrusion prevention, threat intelligence, URL filtering, and more. NGFWs secure outbound traffic from developers to websites like code repositories, east-west traffic between multiple containerized applications or between a containerized application and a legacy application, and inbound traffic that may contain threats.



Comprehensive network security for cloud-native environments requires network anomaly detection, microsegmentation, and firewall protection.

Did You Know?

"Defending the perimeter is no longer an effective strategy. Zero Trust implements methods to localize and isolate threats through microcore, microsegmentation, and deep visibility to give you an organized approach to identify threats and limit the impact of any breach." [8]

[8] "Zero Trust," Forrester, accessed February 18, 2022.

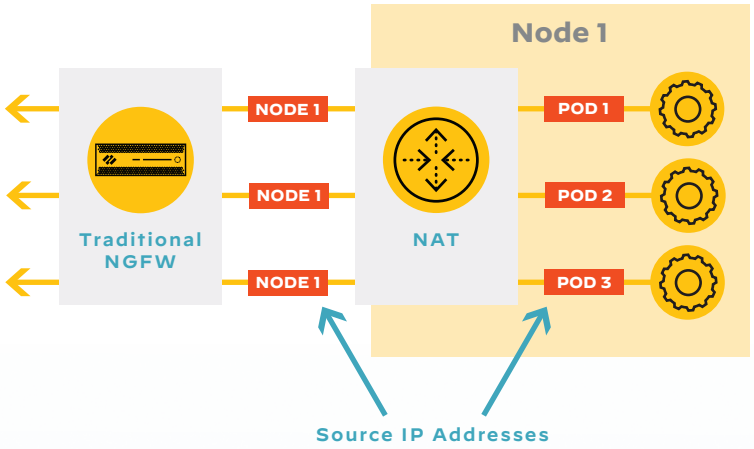
For maximum effectiveness, cloud-native network security needs to leverage native Kubernetes constructs.
Learn more in the next section.

The Need for Container Firewalls

Both physical and virtual NGFWs play an indispensable role in securing on-premises and cloud deployments—few data centers can do without them. However, cloud-native environments pose unique challenges that these kinds of firewall NGFWs were not designed to handle, especially when it comes to looking inside the Kubernetes environment.

In Kubernetes, applications, or **namespaces**, run on **pods** (collections of containers). Pods run on **nodes**, either physical or virtual machines. Developers rarely have to deal with nodes explicitly, but nodes impact how firewalls operate. Because of network address translation (NAT), all outgoing traffic carries the node IP address as the source—the node IP addresses are unavailable. As a result, firewalls sitting outside the Kubernetes cluster are blind to the actual source of the traffic.

For effective security in a container environment, you must know the true source address before NAT. For that reason, the firewall must move inside the kubernetes cluster for maximum effectiveness—the guiding principle behind the Palo Alto Networks CN-Series firewalls.



Due to network address translation (NAT), all outbound traffic carries the node source IP address.

Did You Know?

- Kubernetes organizes containers into **pods**, which are the basic unit for scheduling.
- A **cluster** is a collection of nodes running on the same host. Clusters provide a mechanism for ensuring high availability.
- Depending on the cluster, a **node** is a virtual or physical machine that contains the necessary services pods require.
- A Kubernetes **service** is a set of pods that work together, for example, one tier of a multi-tier application.

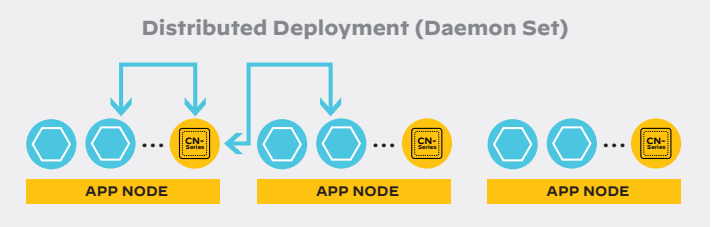
The need for deeper knowledge inside the Kubernetes cluster led to the creation of the Palo Alto Networks CN-Series firewall. Learn more in the next section.

Network Security for Kubernetes: CN-Series from Palo Alto Networks

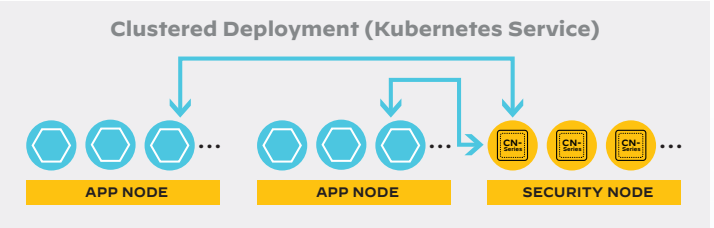
The Palo Alto Networks CN-Series firewall is the industry's first next-generation firewall (NGFW) purpose-built to secure the Kubernetes environment from network-based attacks. With the CN-Series firewall, you gain Layer 7 visibility using native Kubernetes context to protect container-based applications throughout the environment without compromising speed and agility.

The CN-Series can be deployed natively within the Kuberneters environment in two modes. In distributed mode, the firewall dataplane runs as a daemon set on each node. Administrators can deploy firewalls on all cluster nodes with a single command, placing security controls as close to workloads as possible. In clustered deployment mode, the firewall dataplane runs as a Kubernetes service on dedicated/shared security nodes. Clustered deployments are best suited for large Kubernetes environments where a distributed deployment would be resource intensive and cost prohibitive.

In addition, native integration between Panorama and Kubernetes enables CN-Series firewalls to leverage contextual information about containers in the formulation of security policies. For instance, container namespaces can be used to define a traffic source in a firewall policy.



The firewall dataplane runs as a daemon set on each node.



The firewall dataplane runs as a Kubernetes service in a dedicated security node.

Did You Know?

The CN-Series is the containerized form factor of Palo Alto Networks NGFW, which brings to bear the same runtime security as other Palo Alto Network firewalls, including Threat Prevention, DNS Security, URL Filtering, WildFire, and App-ID.

By virtue of its unique design and deployment model, the CN-Series delivers key benefits that are critical for cloud-native application security—read about them next.

How the CN-Series Benefits Security Teams and Developers

Gain Layer 7 visibility and enforcement using native K8S context to protect against known and unknown threats

The CN-Series provides Layer 7 visibility and context into Kubernetes environments by letting users ingest and use namespaces to create security policies governing pod-to-pod, pod-to-cluster, or pod-to-extranet traffic. CN-Series integrates security capabilities directly into the container environment, overcoming the limitations of traditional firewalls to protect against known and unknown threats. As a result, security teams have full traffic visibility, including the ever-elusive source IP of outbound traffic.

Centralized Security Management

CN-Series firewall policies are managed from the same Panorama interface as other ML-powered Palo Alto Networks NGFWs. This provides network security teams with a single console to manage network security for workloads across physical, virtual, and containerized environments.

Deploy and Dynamically Scale Network Security for DevOps Speed and Agility

CN-Series firewalls make use of native Kubernetes orchestration. As a result, DevOps team can use tools and processes they are already familiar with such as Helm charts, YAML files and Terraform templates to integrate CN-Series deployment directly into the CI/CD development process for frictionless deployments.

The CN-Series easily auto-scales for developer needs. When infrastructure grows, traffic increases, or firewall needs expand, organizations can spin up more dataplane pods using Software NGFW credits to scale firewall deployments without compromising DevOps speed.

Protect Containerized Apps Deployed Anywhere

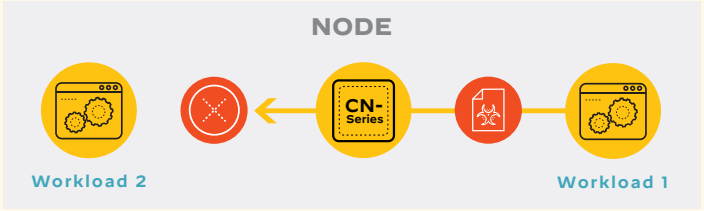
CN-Series is supported on a variety of platforms including Google Kubernetes Engine, Azure Kubernetes Service, Amazon Elastic Kubernetes Service, RedHat Openshift and Tanzu. Now customers have the full flexibility of using the platform of their choice while reaping the benefits of the CN-Series firewall.

The CN-Series is a versatile solution that covers a wide range of use cases, including the four described next.

Typical CN-Series Use Cases

Stop Lateral Movement of Threats

The CN-Series prevents lateral movement of threats from an infected workload to other workloads within the node.

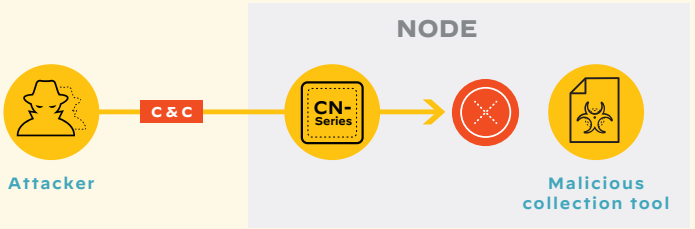


Guard Against Malicious Downloads

The CN-Series limits the allowable access to outside repositories to prevent malicious downloads. In this case, the CI tool can only request specifically allowed information such as Name—all other requests are blocked.

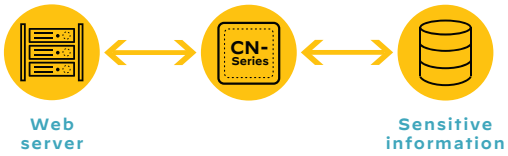
Prevent Data Exfiltration

Even if attackers succeed in penetrating perimeter defenses and installing malicious collection tools, the CN-Series prevents the attacker from communicating, effectively thwarting attempts at data exfiltration.



Support Regulatory Compliance

The CN-Series inspects traffic between web server and the database hosting the sensitive information and thereby, ensures adherence to regulatory compliance standards such as HIPAA and PCI.



No matter what your use case, the CN-Series helps you formulate a strategy for security in cloud-native deployments, as described next.

Formulating Your Strategy for Cloud-Native Network Security

As you develop a comprehensive strategy for securing your cloud-native applications, consider the Palo Alto Networks CN-Series—the industry's first NGFW for Kubernetes. Powered by PAN-OS software, the CN-Series provides the same threat prevention as our hardware and virtual next-generation firewalls.

Here are three good reasons to choose the CN-Series:

Kubernetes Context-Aware Layer 7 Policies

CN-Series natively integrates into Kubernetes to leverage container contextual information such as namespaces to create policies.

Flexible Deployment Models

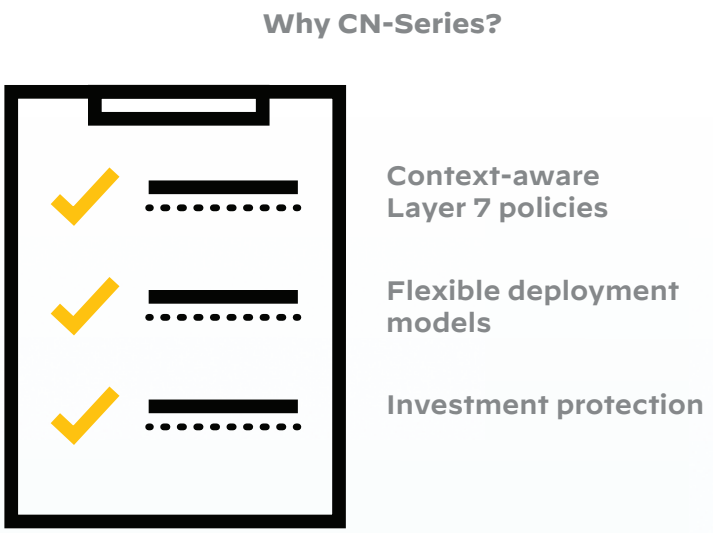
CN-series firewalls offer a range of deployments to meet the needs of both DevOps and NetSec teams.

Investment Protection

You can flexibly use Software NGFW Credits for VM-Series virtual firewalls, CN-Series container firewalls, cloud-delivered security subscriptions, and VM Panorama.

Did You Know?

You can learn more about the CN-Series at our CN-Series webpage.
To see the CN-Series in action, visit QwikLabs Hands-on-Exercise.



paloalto NETWORKS
3000 Tannery Way
Santa Clara, CA 95054
Main: +1.408.753.4000
Sales: +1.866.320.4788
Support: +1.866.898.9087
www.paloaltonetworks.com