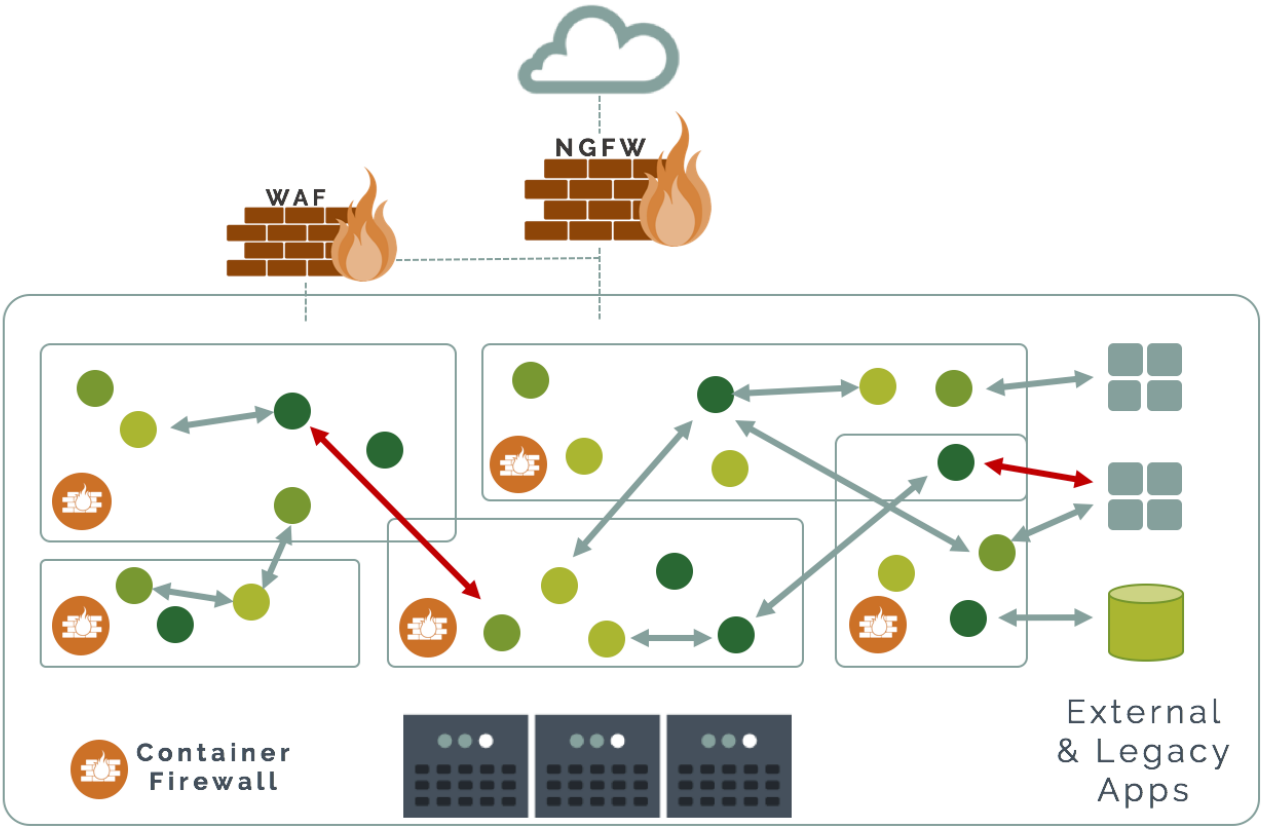
Cloud-Native
Container Firewalls



A Comparison Of
Container Firewalls vs.
Next Generation Firewalls vs.
Web Application Firewalls

NeuVector

Container Firewalls vs. NGFWs vs. WAFs

What is a Container Firewall? And how is it different than a Next Generation Firewall and a Web Application Firewall?

Containers and microservices are revolutionizing computing. But can firewalls help secure these? Next Generation Firewalls (NGFW) were supposed to handle the latest threats and data center designs, but fall short in the new cloud microservices environments. Web Application Firewalls (WAF) provide dedicated protection from malicious HTTP clients attacking web front-ends, but are not designed for internal application protection.

Before we get into the feature comparison of NGFWs, WAFs, and container firewalls, let's take a look at the attributes of containers and microservices. Containers are part of a larger trend toward virtualized application workloads. Virtualized workloads, whether they are containers, IoT devices, or serverless computing provide a wealth of declarative meta-data from which security policies and decisions can be derived.

Attributes of Microservices – An Explosion of East-West Traffic

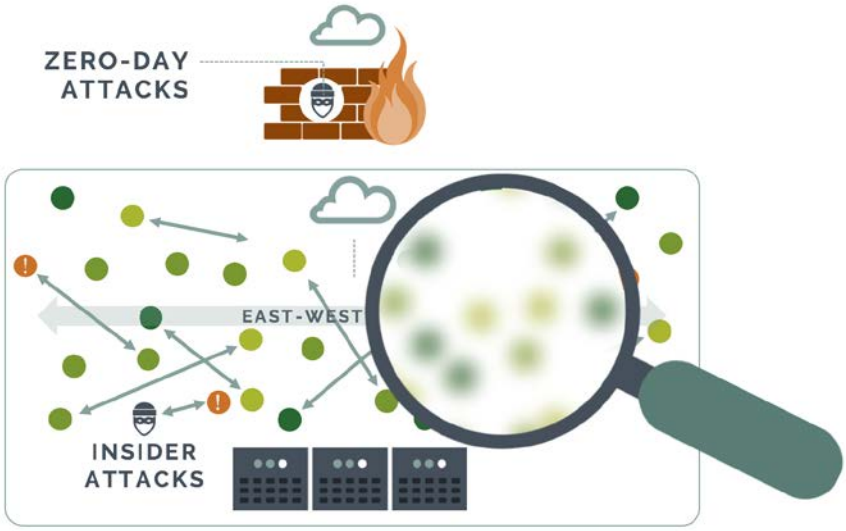
The migration from monolithic applications to container-based microservices brings many benefits but also changes communication patterns. The most significant change from a networking and security view is that there is now an explosion of East-West, or internal, traffic within hosts and between hosts. While each running container can be hardened and expose limited interfaces there are also many more opportunities for attackers to probe and find vulnerabilities.

Containers are designed to be deployed in seconds and an orchestration system can launch new containers on the same hosts or across hosts depending on service demands and host resources available. Each container has its own mapped network interfaces which get assigned and deallocated on the fly.

Security Issues of Container Deployments

With containers being started and stopped constantly, and rapid deployment of updates to applications through a continuous integration and continuous delivery (CI/CD) pipeline, it becomes very difficult to monitor and secure container traffic at the network layer. NGFWs and WAFs are designed mainly to be a gateway for external, or north-south traffic, and can't protect container traffic.

Not only is it difficult for traditional firewalls to see east-west internal traffic within a host or between hosts, it is also impossible for them to keep up with the constant changes as containers launch and disappear. As one network security architect put it "in a containerized world you can't be messing with iptables or manually updating firewall rules."



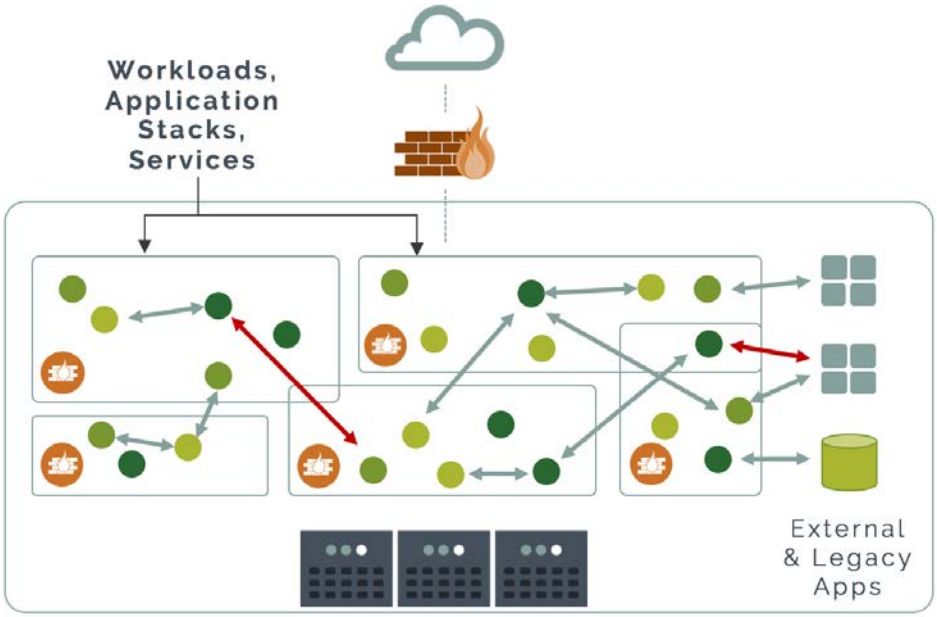
Why is it important to monitor containers at run-time? One reason is the frequent use of open source software for building container applications. Often, developers may not understand the application vulnerabilities which are introduced with each open source package or library used. And once in production, it is easy to lose track of which containers are vulnerable to new vulnerabilities discovered, often years after they are put into production.

Key Features of a Cloud-Native Container Firewall

So what is a container firewall? A container firewall provides much of the same protections that next generation firewalls provide at the edge, but in a cloud-native

environment for all container traffic. This includes east-west, north-south, and container to non-container traffic.

A cloud-native container firewall is able to isolate and protect workloads, application stacks, and services, even as individual containers scale up, down, or across hosts. It must also protect the ingress and egress from external networks and legacy applications much like a traditional gateway firewall does, except with container awareness.



Here are the key features of a Cloud-native Container Firewall

- **Intent based intelligence.** Understands intent of applications from meta-data and behavioral analysis. Characteristics include:
  - Declarative, automated protection. Discovers application behavior and security requirements and adapts to changes and updates.
  - Whitelist based rules. Assumes a zero-trust model and defines allowed behavior.
  - Application based (Layer 7) policy. Does not use IPtables or only L3/L4 rules.
- **Container level protection.** Drop suspicious connections or quarantine entire container.
- **Integrates with container orchestration**. Scales across hosts, clouds and adapts to updates.
- **Supports container platforms and run-time engine.** Runs seamlessly with system security libraries, overlay networks, and Docker engine.

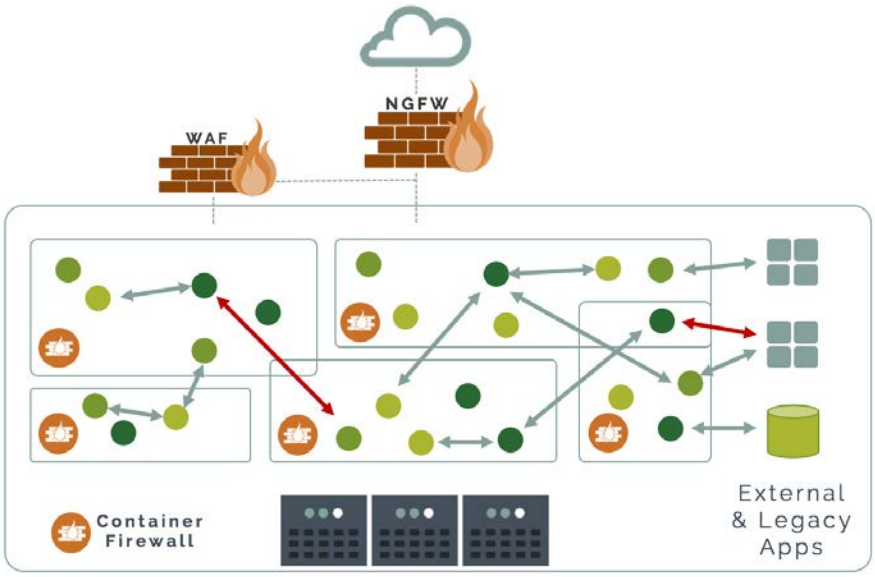
- **Supports common containerized application protocols.** Recognize and enforce policy based on popular application protocols such redis, mysql, mongodb.
- **Fits into CI/CD processes.** Integrate into automated pipelines using REST API to support scripting, Jenkins etc.

A container firewall also includes many next generation firewall features, such as:

- **Layer 7 deep packet inspection (DPI)**. Many microservices communicate over HTTP, and detecting and protecting based on application protocol is critical.
- **Threat protection.** Protects against internal application level attacks such as DDoS, DNS attacks commonly found in web application firewall (WAF) devices.
- **Blacklist rules.** Ability to set rules based on IP addresses, ranges, or other L3/L4 policies.

Because a container firewall is meant to primarily protect container traffic, it is not meant to replace the NGFW, IDS/IPS, or WAF at the edge. However, it must protect against common known application attacks which could originate internally.

The diagram below shows how an NGFW provides multi-protocol protection, while a WAF provides additional web application protection to containers exposed to the internet.



A container firewall provides inspection and protection of all traffic into and out of a container. While this might be primarily container to container traffic, a container

firewall must also protect ingress from external sources to containers, and egress from containers to external applications and the internet.

Key Features of a Next Generation Firewall (NGFW)

Next generation firewalls provide advanced protection for traditional data centers, where traffic from the internet or from untrusted networks need to be secured.

In general, a next generation firewall will include these features:

- **L7 Application Awareness.** Monitor connections by inspecting the application protocols at network layer 7, in addition to Layers 2-4.
- **Threat Detection Through Intrusion Detection/Protection System (IDS/IPS)**. An IPS system will detect and block known attacks through the use of signatures, behaviors, and other detection techniques.
- **Stateful Inspection.** Traffic inspection and protection policies are based on the state of connections. An NGFW can enforce stateful inspection policies based on not only L2-4 but also up to Layer 7.
- **User-Identity Policies.** Access to resources can be controlled by user identity, not just the IP address or type of connection. In a container firewall, a similar capability is to restrict container to container traffic based on behavioral learning.
- **Support for Routed and Bridged Modes**. Firewalls may be deployed as a bridge (L2) and/or a router (L3) depending on the topology and requirements. Container firewalls are host deployed and operate like a bridge (bump in the wire) to monitor packets and block if enabled.

Next Page: NGFW vs Container Firewall

Chart: NGFW vs Container Firewall Comparison

	Next Generation Firewall (NGFW)	Container Firewall
Functions	• L7 Stateful Application Inspection • IDS/IPS • User Identity Policies • Manual blacklist/whitelist policies	• Cloud-native • L7 Inspection / Isolation • Declarative, Automated Policy • Whitelist based • Scale up / down • Internal threat protection • Other container security features*
Deployment	• Gateway/Edge • North-south visibility	• Host • East-west + north-south visibility • Container ingress and egress
Integration	• SIEM	• Container engine (Docker) • Orchestration (e.g. Kubernetes) • CI/CD pipeline • SIEM

**See Continuous Security below for additional container security features*

Key Features of a Web Application Firewall (WAF)

Web application firewalls provide advanced protection for web-based traffic, typically HTTP/S where traffic from the internet first interacts with the 'front-end' of an application. Most WAFs detect a number of application threats including the OWASP Top 10.

In general, a web application firewall will include these features:

- **Detect Application Attacks.** Detect SQL injection, Cross-site scripting (XSS), DDoS, DNS attacks etc.
- **Protocol, Logic, and Object Format Support.** JavaScript, SQL, HTML, XML, JSON, Cookies, etc.
- **Support HTTP and HTTPS**. Some WAFs will terminate SSL connections, while others rely on termination by a load balancer in front.
- **Virtual Patching.** Temporarily 'patch' vulnerabilities with network blacklist policies until application can be patched.

Next Page: WAF vs Container Firewall

Chart: WAF vs Container Firewall Comparison

	Web Application Firewall (WAF)	Container Firewall
Functions	• HTTP/S web server protection • Cross-site scripting (XSS), SQL injection, OWASP Top 10, etc. • Advanced inspection of cookies, form data, URIs • Virtual patching • Cloud based signature updates	• Cloud-native • Application Intelligence & L7 Inspection • Declarative, Automated, Whitelist based Policy • Container threat protection • Other container security features*
Deployment	• Gateway/Edge • North-south visibility	• Host • East-west + north-south visibility • Container ingress and egress
Integration	• SIEM	• Container engine (Docker) • Orchestration (e.g. Kubernetes) • CI/CD pipeline • SIEM

**See below for other container security features*

Continuous Security for Containers

Like any environment, a containerized environment requires a layered security strategy with multiple protection layers. It's critical to build in security throughout the Build, Ship, and Run cycle. For run-time visibility and protection, a container firewall plays a central role.

"A container firewall can provide not only network layer inspection and protection but is also in a unique position to monitor host and container processes and perform security audits."

Because of its distributed nature, host-based container firewalls provide efficient local monitoring and protection. With its integration with the Docker engine and orchestration and container management tools, a container firewall can also provide host and container process inspection, security auditing and testing, and resource monitoring. Container firewalls often contain the following additional features:

- Host process monitoring and security for privilege escalations, suspicious processes and breakouts
- Vulnerability scanning in registries, hosts, and running containers
- Auditing and compliance with CIS Benchmarks security tests
- Packet capture for forensics and debugging.

Securing container deployments requires some new technologies such as container firewalls as well as traditional solutions such as next generation firewalls. For new virtualized workloads, application intelligence, declarative policies, and integration with cloud-native orchestration tools are required to monitor and secure them.

See the NeuVector Container Firewall In Action

To learn more about NeuVector:

- Contact us at info@neuvector.com
- Visit us at https://neuvector.com

Schedule a private demo, or request a free trial. We'll need your Docker hub ID to enable you to download the NeuVector container.