Securing Cloud Workloads in 5 Easy Steps

As organizations transition to microservices in a public cloud, security becomes a bottleneck. Here's how to fast-track your cloud-native application journey.

View original

Originally published by Tigera.

Written by Senthil Nithiyananthan, Tigera.

As organizations transition from monolithic services in traditional data centers to microservices architecture in a public cloud, security becomes a bottleneck and causes delays in achieving business goals. Traditional security paradigms based on perimeter-driven firewalls do not scale for communication between workloads within the cluster and 3rd-party APIs outside the cluster. The traditional paradigm also does not provide granular access controls to the workloads and zero-trust architecture, leaving cloud-native applications with a larger attack surface.

In this blog post, I'll outline an easy 5-step process for fast-tracking your organization's cloud-native application journey by making security a business enabler while mitigating risk.

Step 1: Visibility

Gaining visibility into workload-to-workload communication with all metadata context intact is one of the biggest challenges when it comes to deploying microservices. You can't apply security controls to what you can't see. The traffic is not just flowing from a client to a server in this new cloud-native distributed architecture, but also between namespaces that reside between many nodes, causing flow proliferation. You want a solution that provides a dynamic visualization of all traffic flowing through your network in an easy-to-read UI.

This visualization should allow you to view all the inside and outside (east-west and north-south) connections. Even better if the visualization is packaged in such a way that allows you to double-click into individual flows to see namespaces making connections, destination service, port number, number of flows, permit or deny, and various other information including process names and IDs. This would provide a tailor-made, live view of your cluster and highlight any vulnerabilities and security gaps.

Step 2: Security Policies

After you've seen the security gaps, the next step is to build and apply security policies to have granular access control of the traffic flow in the environment. You want a solution that provides a tier-based policy model with a high level of granularity for identity-aware microsegmentation (5-tuples and other identity-based elements like namespaces).

When you have more than a couple of microservices running in your environment, building security policies by hand can be a complex task. You'll want a solution that offers a policy recommendation engine that you can use to build a base set of policies, and stage them to understand the impact of those security policies in your environment (i.e. preview the flows that will be affected), thus automating the whole process of building security policies.

Step 3: Advanced Security Controls

Building a zero-trust architecture means having advanced security controls enabled as part of the architecture to identify zero-day threats in the environment. Here are a few examples of features that you'll want to look for in a solution:

- FEODO tracker – A FEODO tracker that lists all the command and control IP addresses in the cluster. Even better if you can automatically block those connections from going out of the pods.
- Honeypods – One way to see if any suspicious connections exist in the environment is for security admins to create honeypods to check if any connections are being made to a pod.
- Data-in-transit encryption – You want to be able to easily encrypt all traffic between pods.
- Runtime threat defense – Look for a solution that offers advanced security features like workload-based IDS/IPS, anomaly detection, and WAF.

Step 4: Troubleshooting

One of the challenges that the security team is always battling when it comes to ephemeral services is that issues don't last long enough to troubleshoot. Find a

tool that offers dynamic packet capture capabilities and provides an easy way to schedule the capture and download packet captures (pcap files) so you can identify and fix any issues in the environment.

Step 5: Compliance Reporting

Look for compliance reporting that provides an easy way to keep track of nodes and namespaces that are protected by ingress and egress rules. You want to be able to easily download and share these with the compliance team to fast-track cloud migrations. You'll also want to be able to easily run benchmark reports against CIS standards for Kubernetes.

The goal of this post is to get you started on your Kubernetes journey so you have peace of mind around enabling security for workloads. This is not a comprehensive list of features that you will need, but is a great first step in applying security and moving from DevOps to DevSecOps on day 0. This will help you reduce time to deployment for your applications in a multi-cloud architecture.

Read our cloud-native security guide to discover the 4 C's of cloud-native security and more ways you can improve security for your cloud-native applications.