How to Secure Containerized Applications

The zero-trust security approach is based on the principle that perimeter defenses will eventually be circumvented by a determined attacker and therefore internal users, assets, and applications are not to be trusted just by default. One such implementation of zero trust is to group systems and data into granular microsegments with only authorizing traffic that are part of valid business processes. This prevents the propagation of malware or ransomware that has slipped past the perimeter defenses.

Traditionally, for server and virtual machine-based applications, the policy enforcement point for this strategy has been at the IP address and port level. Implemented well, it can effectively reduce the attack surface and contain the blast radius of any attack that gets through.

Today, enterprise applications are being architected using microservices. They are often delivered in cloud-deployed Docker containers managed using orchestration tools such as Kubernetes and Docker Swarm. The scalability and agility of microservices, as well as cost savings on cloud resources are drivers for this adoption trend.

Microservices must be protected from lateral propagation of malware or ransomware — the so-called "east-west" axis — in a different way, because they communicate in a different way. Microsegmentation policies based on IP addresses and port numbers will not protect microservices, because they communicate using Layer-7 constructs, i.e., Application Programming Interfaces. Policies defined at the IP address level would be useless in creating microsegments for microservices applications because the components that make up a microservice (e.g., pods) do not have fixed addresses, and use one or more overlay networks.

Protecting microservices from exfiltration of data to the external internet — the "north-south" axis — requires a different approach as well. Some microservices may have to connect with an external IP address as part of a normal business process. In traditional non-microservice applications hardware firewalls have been used to enforce egress controls by mapping internal IP addresses to permitted external IP address connections. This approach will not work well in a microservice environment, because a broad range of IP addresses would have to be permitted, opening the enterprise to risk. Instead, egress control must be addressed at the source: the microservice invoking the connection.

The microsegmentation policy enforcement point for traditional applications has been host-based firewalls (in Windows and Linux operating systems), and hardware firewalls. Policies can be instantiated at these points using an agent installed on each endpoint. For microservice applications, communication policy can be implemented by leveraging the previously mentioned orchestration tools. For example, the Istio service mesh for Kubernetes uses a sidecar proxy with each microservice for observability and traffic management. This sidecar can become the policy enforcement point for zero-trust microsegmentation of the microservices.

Organizations may be challenged by the fact that the team which manages network security is separate from the cloud team which manages the containerized microservices applications. The cloud team may not be focused on zero trust security, and the network security team may not understand the fundamental differences in the vulnerabilities of non-IP and port-based communications. A further complication is that containerized microservices often also communicate with traditional non-containerized applications in the enterprise landscape. If microsegmentation policies are not also enforced at the microservice API level, they could introduce an unforeseen vulnerability to the network-defined microsegments.

An ideal approach to bridge this gap would be a solution that allowed a unified approach to zero-trust microsegmentation, one that supports both traditional IP and port-based traffic, as well as API-based microservices communications, in a unified platform that could be leveraged both by the network security team, and the cloud team. This platform would be a centralized policy engine to translate microsegmentation policies into the specific firewall rules for server and VM based applications as well as configuring the sidecar proxy for microservices based applications.

Ideally, the solution would be a holistic approach with a unified administrator user interface that could be leveraged collaboratively between the cloud team and the network security team, and which will ensure that micro-segmentation polices are enforced in the different modes necessary to protect the whole enterprise landscape. Otherwise, an onerous burden would be placed on those responsible for segmentation policy, in that they would need to manually maintain coherence between polices separately enforced on the IP-based and the API-based enforcement points. With such a unified zero trust platform, a coherent and effective policy regime can be deployed and managed to protect the whole landscape, across multiple communication protocols. This strategy would protect the enterprise in the event of a perimeter breach and support multiple stakeholders on both the security team and the cloud team.