NIST 800-190 Application Container Security

Araali Networks

July 2020

Introduction

Modern cloud-native application architectures give developers the ability to test, iterate, and release applications and microservices at the speed of light. Also, companies are adopting hybrid and multi-cloud strategies to lower cost and keep vendors honest. This strategy helps companies build performance and resiliency as it allows for scaling and redundancy on one hand while mitigating cost and downtime risks on the other hand.

However, cloud-native presents its own set of challenges around security and compliance, especially for companies in the regulated industry. Enterprises have to reevaluate and rethink their security processes and security solutions to remain compliant with regulatory controls.

What is NIST 800-190

The National Institute of Standards and Technology (NIST) published a special publication to share recommendations and guidelines addressing potential security concerns associated with the use of containers.

The special publication had a section, section 4, dedicated to the container security countermeasures. In this document, we will highlight how Araali enables DevOps and Security teams to easily bring some of those controls to their deployments.

Countermeasures for Major Risk		Araali
4.1 Image Countermeasures		
	4.1.1 Image vulnerabilities	Araali can look into image vulnerabilities during CI/CD pipeline when the application runs in the CI environment. This simplifies integration steps, and all the containers that make it to CI get scanned even if it came from an unknown/ unsanctioned repository.
	4.1.2 Image configuration defects	Araali can detect modifications to binary and also prevent any unknown binaries from running in prod. Only images that are previously seen and inventoried will run.

	4.1.3 Embedded malware	Araali monitors images at runtime to see if they are doing unexpected networking behavior. It automatically disallows any activities that are not matching its whitelisted policies detected in CI phase. Alerts are automatically routed to the right app team.
	4.1.4 Embedded clear text secrets	Araali gives cryptographic identity to every app. Apps communicate with each other based on these identities and not embedded secrets. Even if clear text secrets are present, they will never get exploited.
	4.1.5 Use of untrusted images	Araali will not allow any image not previously seen and blessed in CI/dev or staging env to run in production env. Araali keeps a hash/cryptographic signature of the image and does not allow any unknown image to run.
4.2 Registry Countermeasures		
	4.2.1 Insecure connections to registries	Araali complements a secured registry. If the registry is run with Araali agent, it can enforce right whitelisted activites on the registry host.
	4.2.2 Stale images in registries	Araali gives an inventory of all images in use. This gives teams a bird eyes view on images that are used.
	4.2.3 Insufficient authentication and authorization restrictions	If the image registry runs on in your environment, Araali can cover the host and apply authentication and authorization in addition to existing rules.
4.3 Orchestrator Countermeasures		
	4.3.1 Unbounded administrative access	Araali creates RBAC only to allow the right app teams to access and configure policies for their apps. One app team cannot look into the apps of another group.
	4.3.2 Unauthorized access	Araali gives an inventory of assets and shows if any unauthorized app tried to access any other apps. This shows up as alerts that are intelligently routed to the app owners. Araali can also create mTLS to prevent MITM

		attack.
	4.3.3 Poorly separated inter-container network traffic	Araali creates out of box segmentation for network traffic based on app boundaries or at a finer service boundary. Araali creates a security overlay, the containerized app might be running across hybrid clouds, but it shows up as a consistent single app with app-level segmentation enabled. Araali automatically discovers policies and creates whitelisted app access control out of the box.
	4.3.4 Mixing of workload sensitivity levels	Araali creates out of box segmentation for network traffic based on app boundaries. This can be used to segment sensitive workloads for less sensitive ones automatically.
	4.3.5 Orchestrator node trust	Araali is based on whitelisted policies. Whenever a node is compromised, it does new things which get detected, and the node can be quarantined. Similarly, if a node with an unknown app tries to join, it cannot authenticate and join the Araali security overlay.
4.4 Container Countermeasures		
	4.4.1 Vulnerabilities within the runtime software	Araali agent runs on every node and future node of k8s cluster to monitor all containers that get deployed. If a new vulnerability is discovered and published, Araali will flag the container. If a vulnerability is exploited (zero-day), it will try to run new processes and do new things, Araali will prevent it.
	4.4.2 Unbounded network access from containers	Araali is a node-based distributed firewall that creates out of the box, whitelisted egress policies for every app. Every container running that app gets the egress policy. The egress policies cover inter-app chatter and traffic going to the internet (DNS, fqdn policies).
	4.4.3 Insecure container runtime configurations	Araali complements runtime configuration by allowing only whitelisted communications to happen.

	4.4.4 App vulnerabilities	Araali creates out of box whitelisted policies for the app. Once Araali sees a new process or anomaly, it automatically blocks it (Enforcement Mode) or raises an alert (Alert Mode). This covers unexpected process execution, system calls, app binary changes, traffic sent to an unexpected destination, or unexpected networking behavior.
	4.4.5 Rogue containers	Araali out of the box creates zones (production, staging, dev) to segment the app and ensure any rogue container from one environment cannot run in another environment.
4.5 Host OS Countermeasures		
	4.5.1 Large attack surface	Araali is based on whitelisted policies. It only allows activities and network connections that are whitelisted, thereby reducing the attack surface.
	4.5.2 Shared kernel	Araali cannot prevent containerized and non-containerized workload from running on the same host. Still, it can out of box segment the apps ensuring that application access control is individually applied and enforced at an application grain.
	4.5.3 Host OS component vulnerabilities	Araali runs on the host to understand and bubble up host OS vulnerability on the assets page.
	4.5.5 Host file system tampering	Araali will not prevent file tampering but will not let a tampered file with unknown binary hash to run.

About Araali

For CISOs and App teams, who care about security and compliance without loss in productivity, Araali Networks is a distributed firewall leveraging application identity and context.

Araali covers any workload - BM, VM, container and serverless, and any app - customer or open-source without re-compile. Unlike traditional firewall and contemporary container security companies, Araali protects any workload and leverages its Infra-agnostic and portable policies, which are not IP but app-identity based. This makes cloud adoption simpler while lowering TCO and improving security posture.