
Cloud-Native
Security
The Nature of the Threat
CONTAINER
JOURNAL

Conventional technical controls such as legacy firewalls and antivirus programs with pre-configured rules cannot keep pace with the ephemeral, temporary nature of containers. Containers running on Kubernetes clusters are at the foundation of cloud-native threat often need to communicate with each other as well as with back-end services where even a small mistake can expose sensitive data and credentials with admin privileges.[1]

Misconfigurations are rife. The interval for correcting misconfiguration errors for most organizations (60%) is **a month or more.**

Roughly a third (32%) of enterprises report **unauthorized access to cloud resources.**

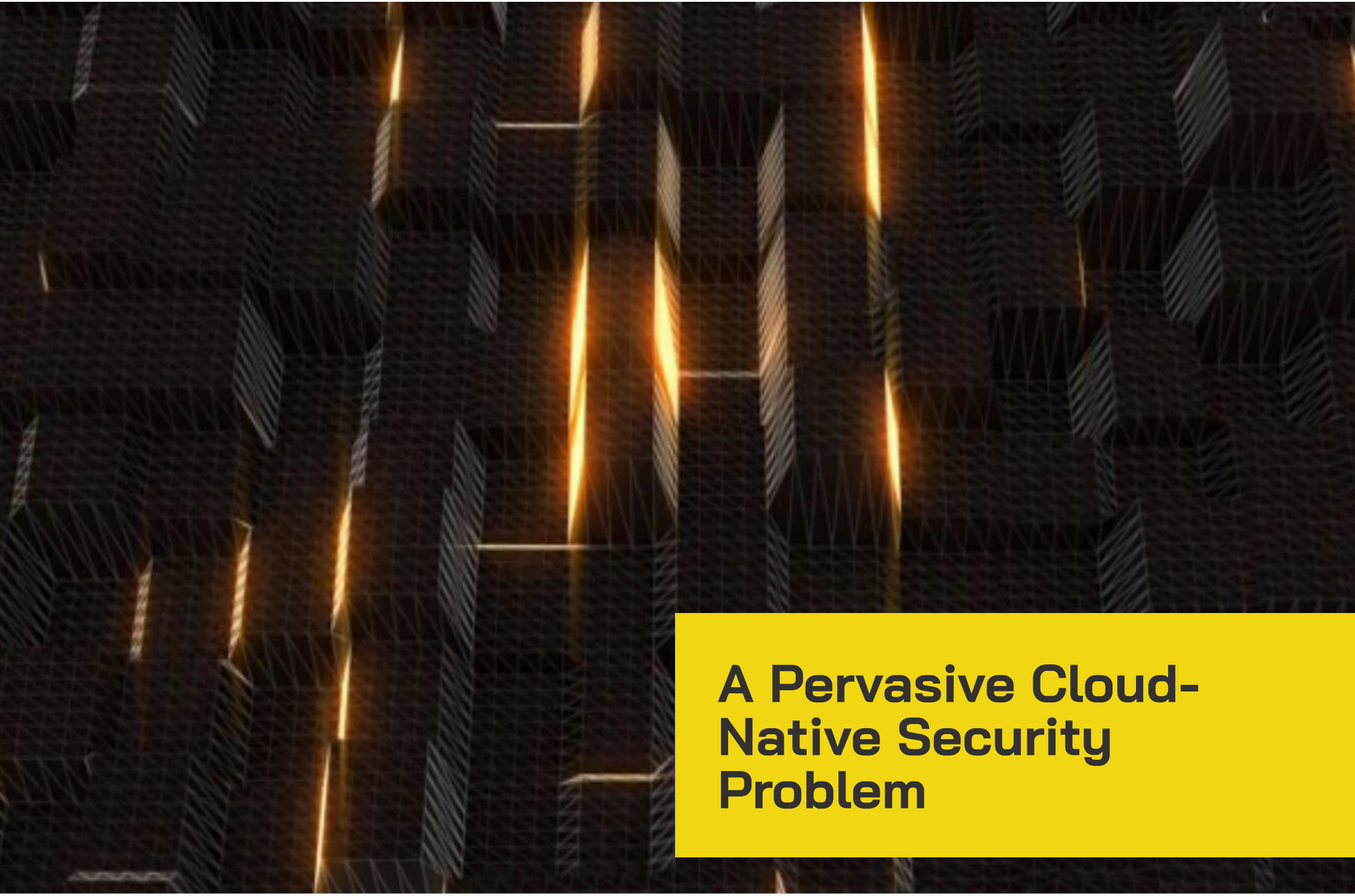
A Growing Cloud-native Security Crisis

It can take less than an hour to exploit vulnerable container infrastructure. Botnets are swiftly finding and infecting new hosts as they become vulnerable, with **50% of misconfigured Docker APIs being attacked within 56 minutes of being set up.**

Attackers have also amplified their use of evasion and obfuscation techniques such as packing the payloads, running malware straight from memory and using rootkits.

FIRST HALF 2020

On average, **daily attacks grew 26% between the first half and second half of 2020,** with 40% of attacks involving either creating backdoors on the host, dropping dedicated malware, creating new users with root privileges and creating SSH keys for remote access.



A Pervasive Cloud-Native Security Problem

A full **94% of organizations have experienced a security incident involving their Kubernetes and container environments during the last 12 months,** with more than half of respondents (55%) needing to delay deploying Kubernetes applications into production because of a security issue.

Just under 60% of respondents also noted there was a misconfiguration incident in their environments over the last 12 months. Nearly half (47%) are still worried about exposures due to misconfigurations in their container and Kubernetes environments.

Overall, nearly a **third of respondents** said they experienced a runtime security incident, while **another third** said they had discovered a major vulnerability.

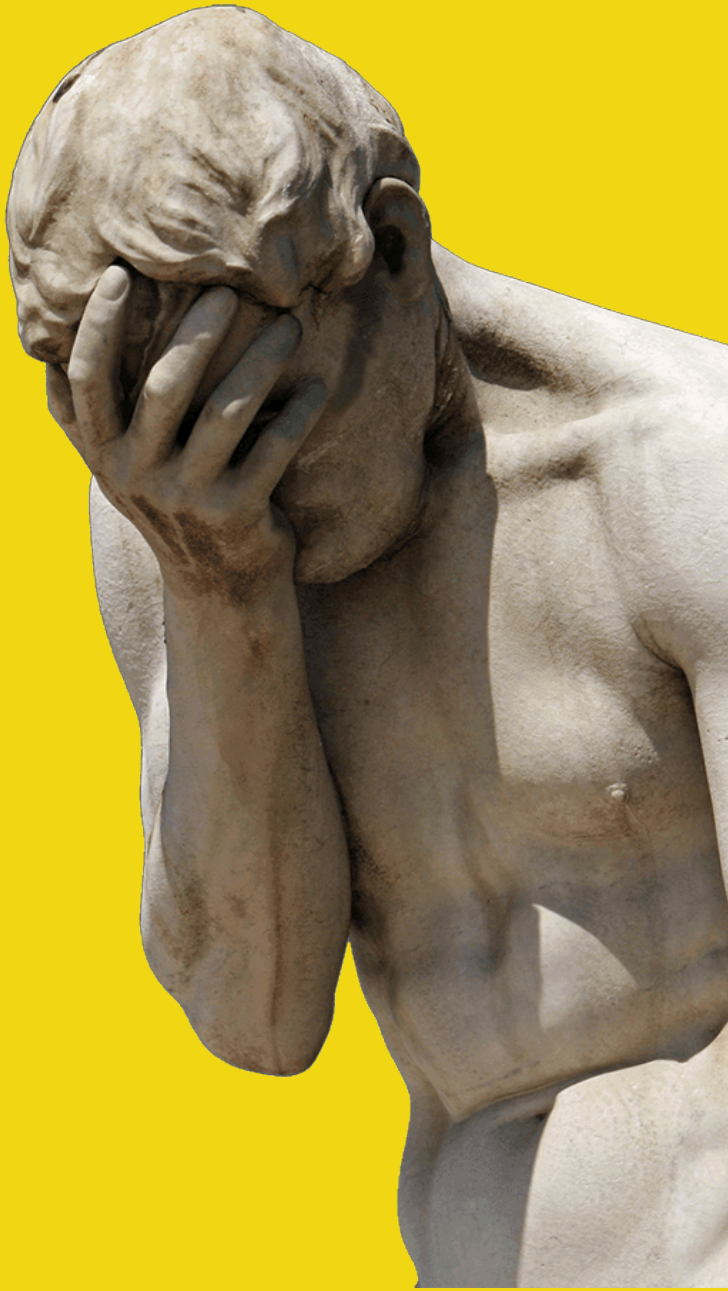
What Makes Investigating Cloud-native Breaches Hard

Containers can be spun up and down within seconds. If an attacker exfiltrates data and then the container is shut down, any record of the attack disappears along with the container.

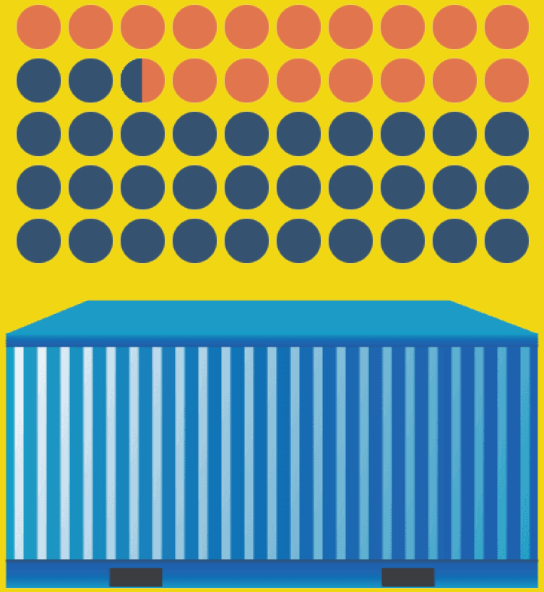
Containers communicate over ad-hoc networks defined by virtualization, with frequently changing internal IP addresses.

Applications and interactions with the infrastructure are separated, with the Kubernetes control plane isolated from workloads. Information on changes to orchestration, access and privileges may be provided by Kubernetes API audits, but there is limited visibility into containers.

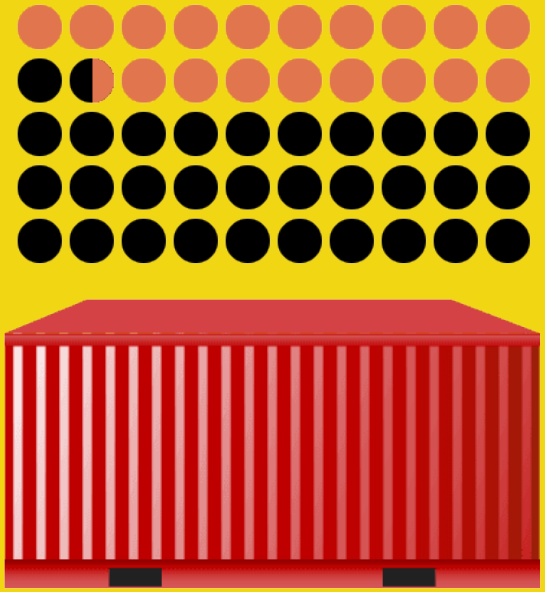
Source



Container Adoption Rises Nevertheless



Nearly **two-thirds of organizations (65%)** now deliver a significant number of applications within containers.



Nearly as many (63%) say increasing the use of containers is their priority.

CONTAINERS INCREASE RISK 28%

CONTAINERS REDUCE RISK 34%

28% of respondents believe containers pose a higher risk. More than a third (34%) believe containers are less risky.

CONTAINER JOURNAL

15 Best Cloud-Native Security Practices

1) Harden Your Operating system to limit the attack surface of container and Kubernetes deployments. Remove all unneeded files and module and apply all security updates and patches.

2) Implement guidelines defined by CIS Benchmark for Kubernetes.

3) Tighten access controls by employing the principle of least-privilege. Customize all permissions to limit access as much as possible.

4) Run vulnerability scanning of containers in the pipeline and in all registries.

5) Run integrity checks to make sure container images are signed. Compliance checks on images should include CIS benchmarks, secrets inspection and other potential image violations.

6) Make sure there are traditional firewalls and other tools to secure the perimeter. Even advanced container and Kubernetes environments must defend against traditional external attacks.

7) Keep container lifespans as short as possible to limit amount of time they might be infected.

8) Load application containers in read-only/non-persistent mode to prevent write access to container images.

9) Isolate or segment running containers to reduce attack surfaces and limit breach blast radius.

10) Monitor attacks in real-time to identify threats at the application layer such as SQL injection attacks.

11) Whitelist process and file activity and map communication among containers to identify anomalous behavior and automatically block unauthorized container access based on abnormal behavior.

12) Run live scans of all running containers to secure a container image as it spins up.

13) Automate security policies in a way that protects both container and the hosts they run on.

14) Avoid running privileged containers. The majority of applications don't require root access to operate, except for system containers like monitoring or logging agents.

15) Analyze security events offline to identify repeated and coordinated attack patterns.

Summary

Cloud-native applications are no more or less secure than any other type of application environment. They are just different. As such, they need to be secured in a way most organizations today have yet to master.

However, as best DevSecOps practices continues to shift responsibility for application security further left toward developers there will come a day when cloud-native applications will be a lot more secure than they are, unfortunately, typically today.

Thank you for reading

Cloud-Native Security