Container vs virtual machine?

A virtual machine runs a full guest operating system on virtual hardware, separated from the host by a hypervisor, a hard boundary. A container runs as an isolated process that shares the host kernel, separated only by Linux features (namespaces, cgroups, capabilities). So a VM is heavier but strongly isolated, while a container is lightweight but its isolation depends on configuration. That shared kernel is why a misconfigured container can be escaped to the host, a risk a VM does not have in the same way.

HOW IT WORKS

01 Why it matters for security

The shared kernel changes the threat model:

- A container escape to the host is possible when isolation is misconfigured (privileged mode-, mounted socket, shared namespaces). There is no hypervisor as a backstop.
- A kernel vulnerability affects every container on the host at once, because they all use that kernel.
- VMs trade performance for a stronger, hardware-enforced boundary, which is why sensitive multi-tenant workloads sometimes run each tenant in its own VM or a sandboxed runtime.

Documented for defensive context.

> WHERE THE WALL IS
>
> *VM: the wall is the hypervisor (hardware-enforced). Container: the wall is kernel configuration (namespaces, capabilities). Softer wall, faster startup, more reliance on getting the config right.*

02 Practical takeaways

- Treat container isolation as configuration, not a guarantee: drop capabilities, no privileged mode, no risky mounts.
- Keep the host kernel patched, since one kernel backs every container.
- For strong multi-tenant isolation, consider a sandboxed container runtime or one VM per tenant.
- Use VMs as the outer boundary and containers inside them, a common and sensible layering.

SOURCES

[1] NIST SP 800-190 Application Container Security Guide

[2] MITRE ATT&CK: Containers Matrix

- Test the actual boundary rather than assuming containers are as isolated as VMs.

Find the container escape paths before an attacker does.

securelayer7.net/learn/containers/what-is-a-container-vs-a-virtual-machine

Open online