What is Docker image security?

Docker image security is making sure the image a container runs from is trustworthy: free of baked-in secrets, free of known-vulnerable packages, built from a trusted base image, and verified before it runs. Images are layered and immutable, so a secret added in one layer stays recoverable even if a later layer deletes it. Weak image hygiene gives attackers credentials, a vulnerable foothold, or a fully poisoned image. Defenses are scanning, minimal trusted bases, no embedded secrets, and signature verification.

HOW IT WORKS

01 What attackers exploit

Image-level footholds attackers look for:

- Secrets in layers: API keys, cloud credentials, or private keys baked in during build and recoverable from layer history even if "deleted".
- Vulnerable packages: outdated OS or app dependencies in the image giving a known exploit.
- Untrusted or typosquatted base images: a poisoned public base that ships a backdoor or miner.
- `latest` and unpinned tags: pulling a mutable tag so the running image silently changes.

Documented techniques shown for defenders.

> LAYERS NEVER FORGET
>
> *Deleting a secret in a later layer does not remove it, the earlier layer still holds it. Never* `COPY` *a secret into a build; use build secrets or runtime injection instead.*

HOW TO DEFEND

- Never bake secrets into images. Use build-time secret mounts or inject at runtime; scan images for leaked credentials.
- Scan images for vulnerabilities in CI and block on criticals.
- Use minimal, trusted base images (distroless or slim) from known registries; pin by digest, not `latest`.
- Verify signatures (image signing) and use a trusted internal registry.
- Rebuild and re-scan regularly so patched packages reach production.

SOURCES

[1] Docker docs: Build secrets

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

[3] MITRE ATT&CK: Containers Matrix

Find the container escape paths before an attacker does.

securelayer7.net/learn/containers/what-is-docker-image-security

Open online