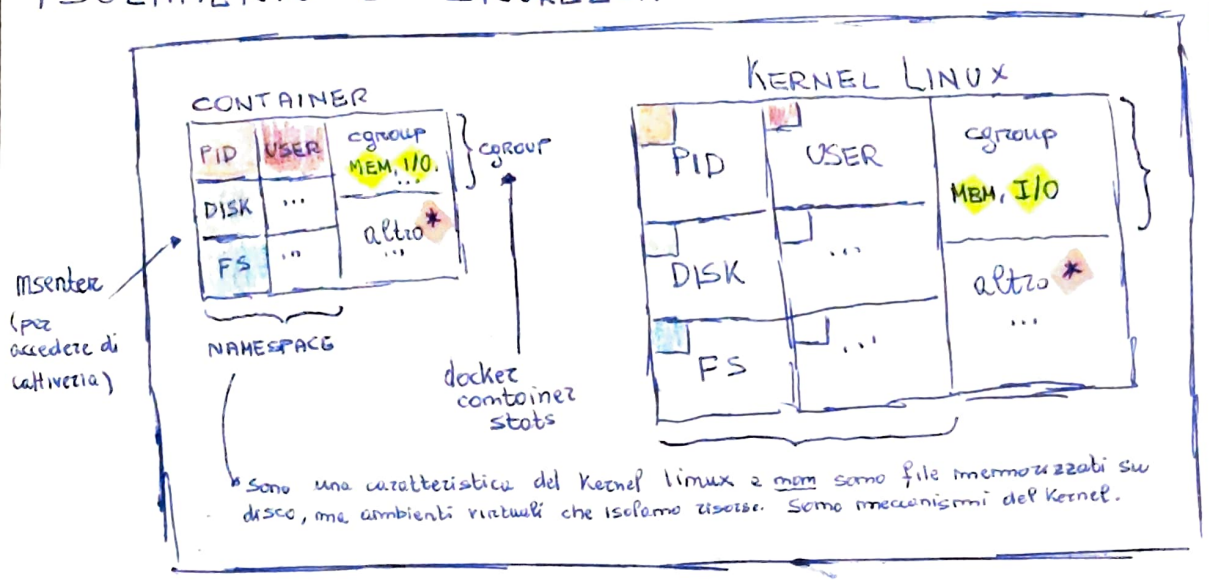
# ISOLAMENTO E SICUREZZA

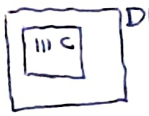



I NAMESPACE garantiscono l'isolamento del container dal sistema host, i CGROUP regolano le risorse che il container può usare.

* Non tutto è isolato tramite NAMESPACE, infatti qui (*) è dove è possibile ci siano problemi legati alla sicurezza.

## RISCHI

### 1) UID 0 ≈ root



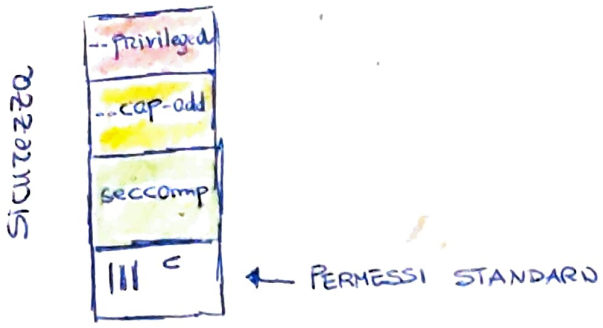

UID 0 in un container `--privileged=true` ≈ UID 0 = root sul DH

- Non usare mai UID = root nel container
- docker container run `-u xxx`
- Aggiungere USER non privilegiato al Dockerfile. (es: USER 500:500)

### 2) ROOTLESS

- Servono delle attivazioni specifiche per usare questa modalità.
- Serve per fare sì che il container non abbia più privilegi dell'utente che lo ha lanciato... anche se dentro al container sono root!
- Utile nei sistemi multi-utente, ma bisogna attivare il LINGER!
- Ci sono però alcune limitazioni generali che possono causare dei problemi

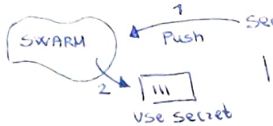
# 3) PRIVILEGI




Il `--privileged=true` permette di avere funzionalità speciali normalmente negate a un container. Espande i privilegi di un container che può essere pericoloso.

Con il `--cap-add=xxx` espandiamo sì i privilegi ma li limitiamo solo a ciò che ci serve.
Abbiamo un controllo più raffinato.
Tuttavia, per esempio se uso `--cap-add=SYS_ADMIN` per concedere la possibilità di fare "umount" in realtà sto concedendo anche molto altro!

Con seccomp posso veramente controllare le singole chiamate di un comando LINUX al Kernel!
Si configura mediante un file .json.




I secret sono
- CRIPTATI
- Messi nel container "in memoria"
- Usabili solo se espressamente esplicitato

Docker

- SECRETS
- DCT — DOCKER_CONTENT_TRUST=1 docker image pull ... (immagini firmate)
- IMG SCAN — Tool di terze parti che cercano vulnerabilità nelle immagini
- SWARM — Sicurezza per costruzione.

Linux

- SECCOMP — Limita le chiamate che possono essere fatte al Kernel Linux.
- MAC — MANDATORY ACCESS CONTROL SYSTEM
  AppArmor/SELinux in base al sistema Linux vengono applicate ai container
- CAP — CAPABILITIES:
  Usare root nel container può essere pericoloso, non usarlo può essere problematico!
  - Però possiamo aggiungere root-powers a un utente.
  - Oppure togliere powers all'utente root
  Anche se scegliere le giuste capabilities può essere difficile.
- CGROUPS — setting limits
  - Di modo che un singolo container non consumi tutte le risorse.
- NAMESPACES

- Ai container, l'isolamento con namespace è automatico