Daemon Security

Docker Daemon Security

El CLI se comunica con un servicio ***daemond*** para crear imágenes o iniciar contenedores. Es fundamental securizar este servicio: cualquiera con acceso a él puede ejecutar comandos Docker en su host.

1. No exponer el socket del demonio Docker.
2. Utiliza TLS si debes exponer el socket daemon.
3. Habilitar el modo rootless.
4. Mantener Docker actualizado.
5. Desactivar la comunicación entre contenedores.
6. Habilitar "*user namespace remapping*".

1. No exponer el socket del demonio Docker.

- El demonio Docker se expone normalmente a través de un socket Unix en /var/run/docker.sock.
- Se puede configurar para que escuche también en un socket TCP, permitiendo conexiones remotas.
- Mantenga TCP desactivado a menos que su caso de uso requiera acceso remoto.

2. Utiliza TLS si debes exponer el socket daemon

- Cuando no hay alternativa al uso de TCP, es esencial proteger el socket con TLS.

- Esto asegurará que sólo se conceda acceso a los clientes que presenten la clave de certificado correcta.

- TCP con TLS sigue siendo un riesgo potencial:
  - Puerto 2376.
  - Certificados autofirmados.
  - TLS 1.0 o TLS 1.1.
  - Robo de claves privadas.

3. Habilitar el modo rootless

- Docker ejecuta por defecto tanto el demonio como los contenedores como root:

  vulnerabilidades en el demonio = vulnerabilidad en tu host.

- El modo Rootless permite iniciar el demonio Docker sin usar root.

5. Desactivar la comunicación entre contenedores.

Docker permite la comunicación entre los contenedores que se ejecutan en su host:

- Automáticamente **TODOS** los contenedores se añaden a la red docker0.

- Esto se llama ICC (Inter-Container Comunication).

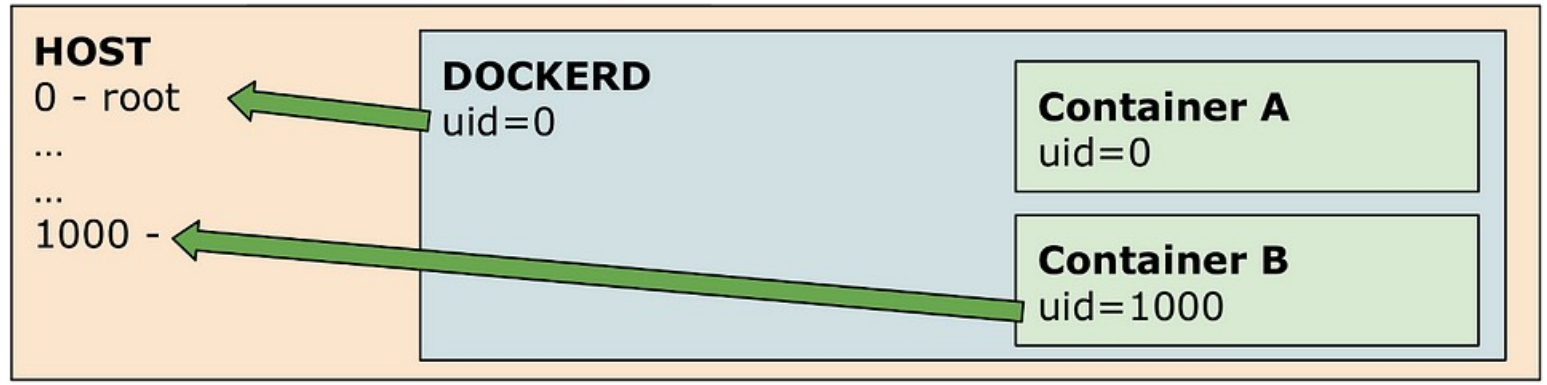
- Se puede deshabilitar y permitir comunicaciones entre contenedores específicos creando redes manualmente.

6. Habilitar “*user namespace remapping*”.

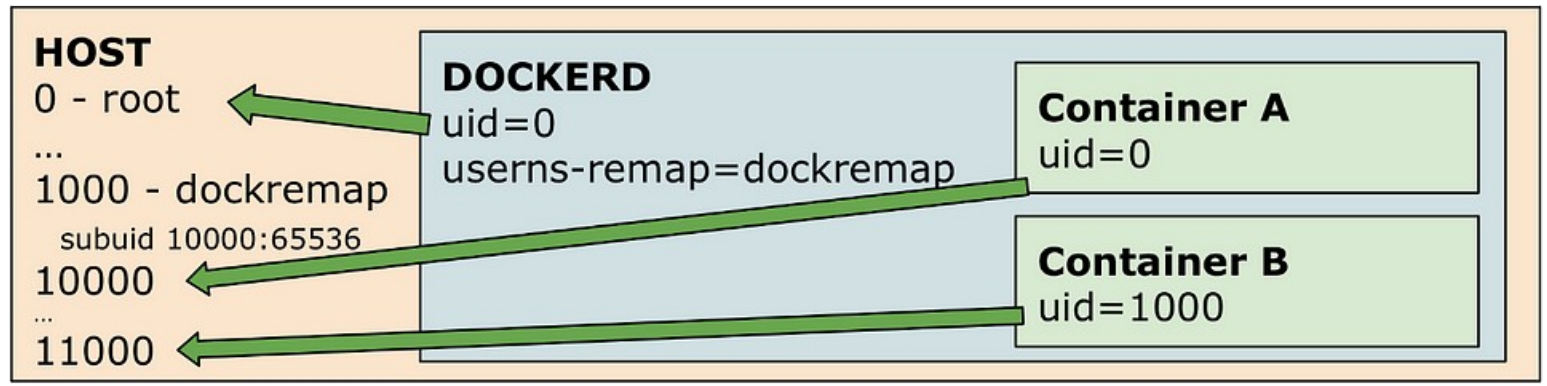
- Función que convierte los UID del host a un rango sin privilegios dentro de tus contenedores.

- Ayuda a prevenir ataques de escalada de privilegios, donde un proceso que se ejecuta en un contenedor obtiene los mismos privilegios que su UID tiene en su host.

No UserNS



With UserNS



Rootless