NTT

Rootless Containers

Akihiro Suda (NTT)
akihiro.suda.cz@hco.ntt.co.jp

Rootless containers

NTT

- Puts container runtimes (as well as containers) in a user namespace
  - UserNS: Linux kernel's feature that maps a non-root user to a fake root (the root privilege is limited inside the namespace)
- Can mitigate potential vulnerabilities of the runtimes
  - No access to read/write other users' files
  - No access to modify the kernel
  - No access to modify the firmware
  - No ARP spoofing
  - No DNS spoofing

e.g., runc breakout CVE-2024-21626 (2024-01-31)

- Also useful for shared hosts (High-performance Computing, etc.)
  - Works with GPU too

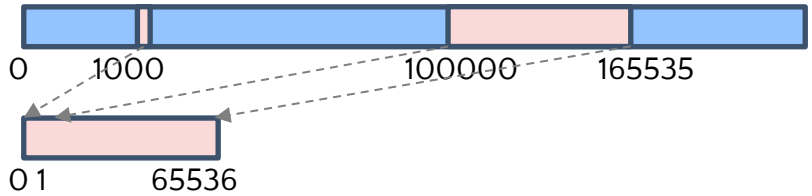
History

- **2014**: LXC v1.0 introduced support for Rootless containers (called “unprivileged containers” at that time)
  - Networking depends on a SETUID binary, which is hard to configure and also is insecure
- **2016**: Singularity v2.2 gained initial support for Rootless
- **2017**: runc v1.0-rc4 gained initial support for Rootless
- **2018**: Several works has begun to support Rootless in containerd, BuildKit, Docker, Podman, etc.
  - slirp4netns (usermode TCP/IP) eliminated the need to use a SETUID binary for bringing up container-to-container networks
- **2019**: Docker v19.03 was released with an experimental Rootless support
- **2020**: Docker v20.10 was released with general availability of Rootless

User namespaces

- Linux kernel's feature to remap UIDs and GIDs

```
# /etc/subuid
1000:100000:65536
```



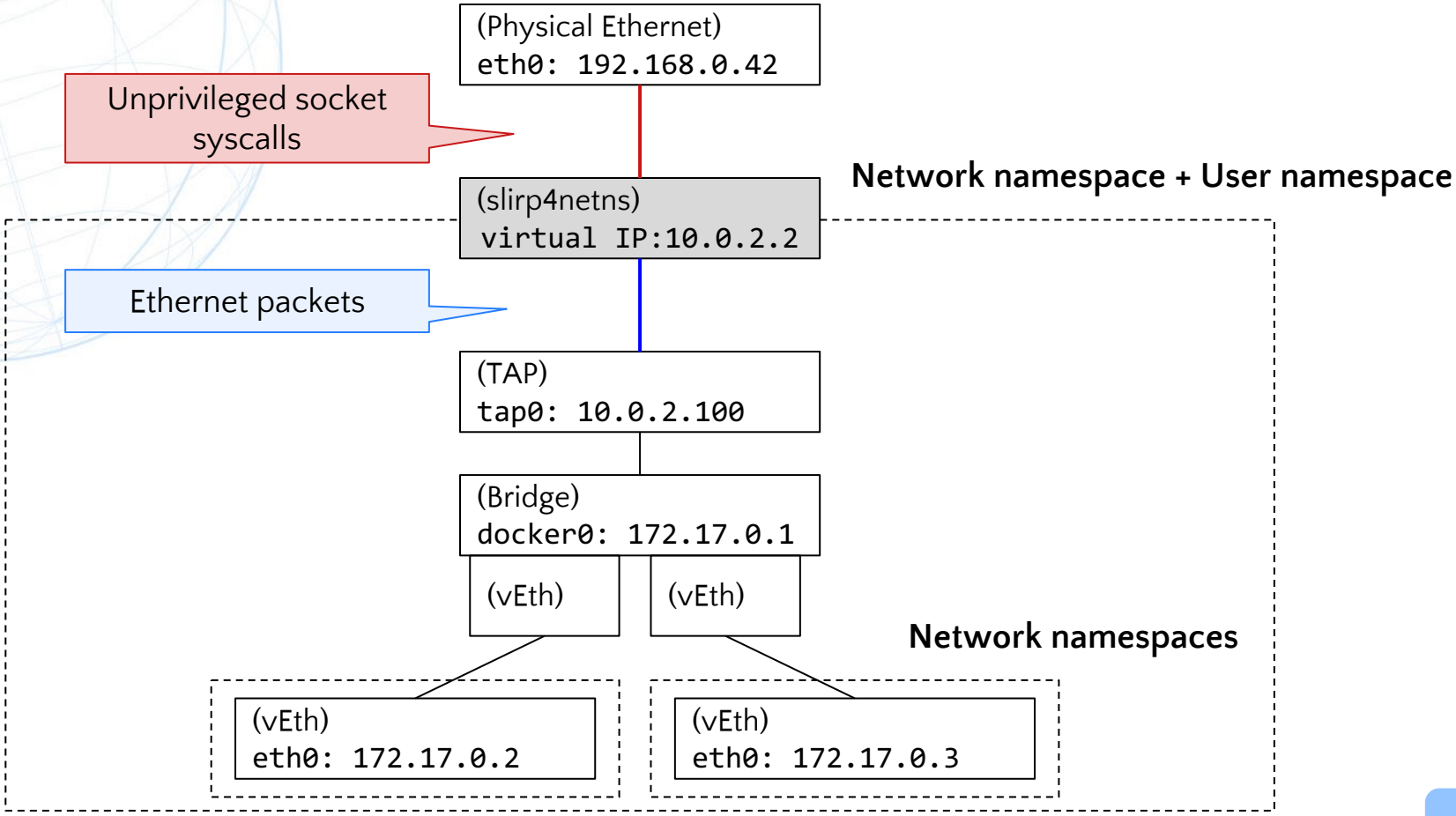
  - UID=1000 gains **fake** root privileges (UID=0) that are enough to create containers
  - The privileges are limited inside the namespace

- Typically at least 65,536 subuids have to be allocated for containers
  - Static configuration (`/etc/subuid`):
    most common, but can be a mess for shared computing
  - Dynamic configuration (`nsswitch`):
    more preferrable for shared computing
    - e.g., via FreeIPA https://freeipa.readthedocs.io/en/latest/designs/subordinate-ids.html

User namespaces

- POC of subuid-less rootless containers is also available, but not ready to be used yet
  https://github.com/rootless-containers/subuidless
  - Emulates UID-related syscalls such as `chown(2)` using `seccomp_unotify(2)` and `xattr(7)`
  - More syscalls have to be emulated

Networking stack



Faster networking (for runtimes)

NTT

- Rootless Docker daemon is executed in slirp4netns's NetNS too, for ease of implementation
  - Slow pull/push
  - No direct access to localhost registries
  - No support for --net=host

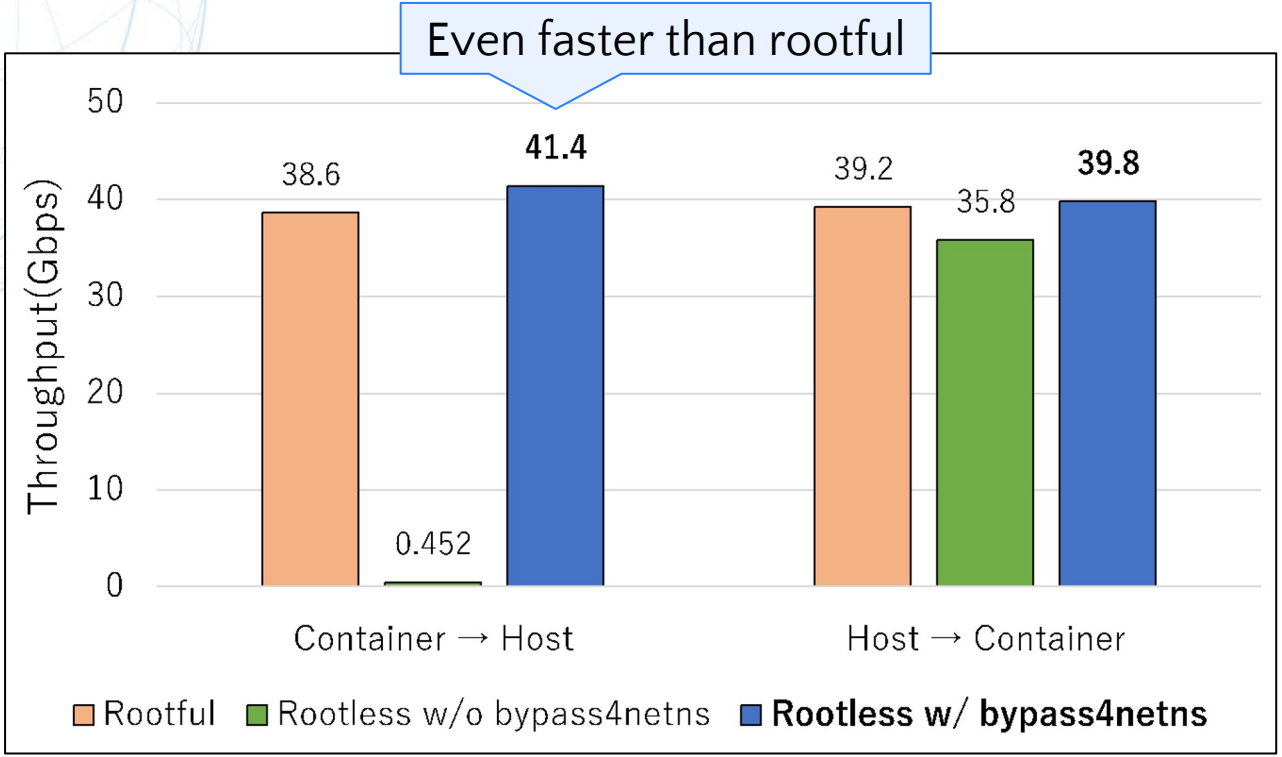
- Docker v26 (or later) may execute the daemon outside slirp4netns's NetNS to eliminate the restrictions 🎉
  https://github.com/moby/moby/pull/47103 (WIP)

- The same technique has been used by Podman and nerdctl (contaiNERD CTL) v2 too

Faster networking (for containers)

- Bypass4netns allows bypassing slirp4netns
  https://github.com/rootless-containers/bypass4netns
- Captures socket syscalls inside the NetNS, reconstructs the FDs outside the NetNS, and replaces the FDs inside the NetNS
- Integrated into nerdctl (opt-in)
- Can be used with Docker and Podman too

Faster networking (for containers)



Accelerating TCP/IP Communications in Rootless Containers by Socket Switching (Naoki Matsumoto and Akihiro Suda, SWoPP 2022)
https://speakerdeck.com/mt2naoki/ip-communications-in-rootless-containers-by-socket-switching?slide=4

Criticisms against Rootless containers (and solutions)

NTT

- It is controversial whether non-root users should be allowed to create user namespaces

- **Yes**, for container users, because rootless containers are much safer than running everything as the root

- **No**, for others, because it can be rather an attack surface
  CVE-2023-32233: Privilege escalation in Linux Kernel due to a Netfilter nf_tables vulnerability

- Several mechanisms are being worked on to conditionally enable unprivileged user namespaces

Criticisms against Rootless containers (and solutions)

NTT

- Linux v6.1 (2022) introduced a new LSM hook: `userns_create`
  - Hookable from KRSI (eBPF LSM)
  - Userspace tools have to be improved to provide a human-friendly UX for this

- Ubuntu 23.10 introduced a new sysctl value `kernel.apparmor_restrict_unprivileged_userns`
  - `/etc/apparmor.d/usr.bin.<FOO>` profile is needed to create UserNS
  - Older releases of Ubuntu were using `kernel.unprivileged_userns_clone` (system-wide single boolean value)

LSM: Linux Security Module, KRSI: Kernel Runtime Security Instrumentation

Rootless Kubernetes

Rootless Kubernetes

- Usernetes: Rootless Kubernetes
  https://github.com/rootless-containers/usernetes

- The current version is implemented by running Kubernetes inside Rootless Docker/Podman/nerdctl

- Multi-node networking is possible with VXLAN (Flannel)

History

- Began in 2018
  - As old as Rootless Docker (pre-release at that time) and Rootless Podman
- The changes to Kubernetes was merged in Kubernetes v1.22 (Aug 2021)
  - Feature gate: `KubeletInUsernameSpace` (Alpha)
- The feature gate is also adopted by:
  - kind (with Rootless Docker or Rootless Podman)
  - Minikube (with Rootless Docker or Rootless Podman)
  - k3s

Usernetes Gen 1 vs Gen 2

NTT

"The hard way"

Similar to `kind` and minikube, but supports real multi-node

	Gen 1 (2018-2023)	Gen 2 (2023-)
Host dependency	RootlessKit	Rootless Docker, Rootless Podman, or Rootless nerdctl (contaiNERD CTL)
Supports kubeadm	No	Yes
Supports multi-node	Yes, but practically No, due to complexity	Yes
Supports hostPath volumes	Yes	Yes, for most paths, but needs an extra config

Usage

```
# Bootstrap the first node
make up
make kubeadm-init
make install-flannel
```

```
# Enable kubectl
make kubeconfig
export KUBECONFIG=$(pwd)/kubeconfig
kubectl get pods -A
```

```
# Multi-node
make join-command
scp join-command another-host:~/usernetes
ssh another-host make -C ~/usernetes up kubeadm-join
```