New Container Kernel Features

Christian Brauner
LXD maintainer & Kernel engineer
@brau_ner
https://brauner.io
christian.brauner@ubuntu.com
Seccomp: notify target

- Allows running less privileged containers
  Unprivileged containers can be granted very specific privileges.

- Seccomp asks userspace for return value and errno
  Execution does NOT continue in the kernel, userspace must do the work.

- Initial support landed in 5.0
  Userspace requires un-released libseccomp.
Builds on top of existing notify target
Effectively a new type of return value from userspace.

Allows for complex userspace filtering
For cases where the kernel cannot filter on some arguments.

No raised privileges
Execution continues in the kernel with original privileges.
Seccomp: extended syscall filtering

- **In-kernel filtering of pointer arguments**
  
  *Filter syscalls such as clone3(), bpf() etc.*

- **Discussion scheduled for KSummit in Lisbon**
  
LSM: stacking

- **Run Ubuntu on Fedora (AppArmor) or Android on Ubuntu (SELinux)**
  Allows them to retain their individual LSM policies.

- **Goal is to stack major LSMs on top of each other**
  AppArmor on SELinux or SELinux on AppArmor.

- **Currently can stack minor LSMs with major LSMs**
  TOMOYO, loadpin, etc. with AppArmor or selinux.
LSM: safeSetID

- **Restrict id transitions through setid-like syscalls**
  
  *System policy determines what transitions are allowed.*

- **Mostly useful for privileged containers**

  *Can be used to allow a limited range of uid/gid for the container.*

- **Will be in Linux 5.3**
New mount API

- **Use file descriptors for mounting**
  Mounts are created, configured, and setup via file descriptors.

- **Anonymous mounts**
  Mounts that are not attached to any path in the filesystem.

- **Avoids numerous race conditions**
  Container managers cannot trust the container’s mount table.

- **Potential for clean uid/gid shifting**
  Shiftfs-in-vfs approach.

- **Potential for setting up namespace**
  Mounting into a set of namespaces.
Keyring namespacing

- **Namespace keyring facility**
  
  *Allows to have per-container keyrings.*

- **Use by network filesystems**
  
  *Keyring namespaces will allow per-container authentication.*
openat2() syscall

- **Restrict path resolution**
  
  \[\textit{LOOKUP\_NO\_XDEV, LOOKUP\_NO\_MAGICLINKS,}
  \textit{LOOKUP\_BENEATH, LOOKUP\_NO\_SYMLINKS, LOOKUP\_IN\_ROOT}\]

- **Restrict O\_PATH file descriptors**

  \textit{Prevent elevating permissions through magic symlinks.}
pidfd API

- **File descriptor referring to a process**
  
  Eliminates inherent races in process management.

- **Get with clone() and CLONE_PIDFD**
  
  Request pidfd be returned together with pid.

- **Send signals with pidfd_send_signal()**
  
  Race-free signal sending (no accidental wrong target).

- **Get pidfd for an existing process with pidfd_open()**
  
  Create pidfd for processes created without CLONE_PIDFD.

- **Poll on a pidfd**
  
  Get exit notification for non-child processes.
- **Request a specific PID**
  *Process will be created with a specific PID.*

- **Interesting for CRIU**
  *Significantly improves restoring of container workloads.*
Questions?

Christian Brauner
LXD maintainer & Kernel engineer

@brau_ner
https://brauner.io
christian.brauner@ubuntu.com