CRIU and the PID dance

Adrian Reber

Linux Plumbers Conference 2019
September 10

Checkpoint/Restore in Userspace

CRIU

CRIU

As transparent as possible

As transparent as possible

PID stays the same

PID stays the same

Independent of PID namespaces

CRIU morphs itself into the to be restored process

`clone()` for each PID/TID

`clone()` for each PID/TID

PID dance

PID dance

`open()` `/proc/sys/kernel/ns_last_pid`

`write()` (PID - 1) to `ns_last_pid`

`close()` `ns_last_pid`

`clone()`

`getpid()`

Avoiding the PID dance (2010):

`eclone()`

`https://lore.kernel.org/patchwork/patch/198220/`

Avoiding the PID dance (2019):

`clone3()`

"In general, clone3() is extensible and allows for the implementation of new features."

`https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=7f192e3cd316ba58c`

clone3() with set_tid

```
1         struct clone_args args = {0};
2         args.set_tid = 2019;
3         syscall(__NR_clone3, args, sizeof(struct clone_args));
```

Relax CAP_SYS_ADMIN

Relax CAP_SYS_ADMIN

Rootless Container Migration

Relax CAP_SYS_ADMIN

MPI Checkpoint/Restore

Relax CAP_SYS_ADMIN

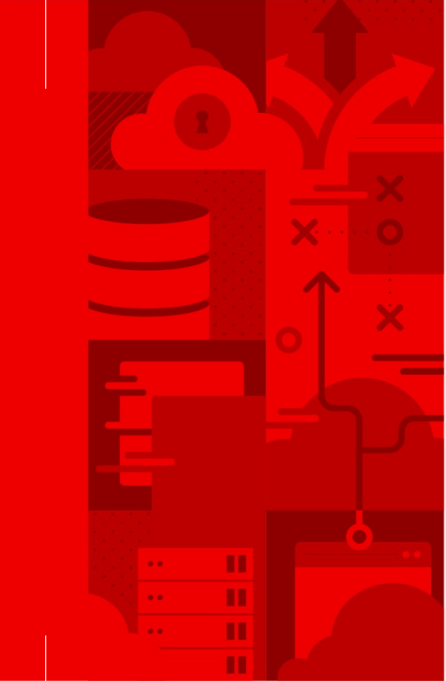
CAP_RESTORE

https://lisas.de/~adrian/criu-and-the-pid-dance-article.pdf

https://lore.kernel.org/patchwork/patch/198220/

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=7f192e3cd316ba58c

https://twitter.com/adrian__reber



Thank you

Red Hat