

Syd+Youki=Syd-OCI

Introduction to a Secure Container Runtime for Linux

Ali Polatel



FOSDEM, 2025

Before we start...

The game is on! Viva la revolución!

- CTF: {https,ssh}://syd.chesswob.org
 - user/pass: syd
 - rules: /etc/user.syd-3
 - goal: read /etc/CTF & get 200€!
- GIT: <https://gitlab.exherbo.org/sydbox/sydbox.git>
- DOC: <https://man.exherbolinux.org>
- ML: <https://lists.sr.ht/~alip/exherbo-dev>
- IRC: #sydbox at Libera
- Matrix: #sydbox:mailstation.de

History: Youki.git/README.md

All you touch and all you see is all your life will ever be.

- Implementation of the OCI runtime-spec in Rust, similar to runc.
- `youki` is pronounced as /jouki/ or yoh-key. `youki` is named after the Japanese word '`youki`', which means 'a container'. In Japanese language, `youki` also means 'cheerful', 'merry', or 'hilarious'.
- Here is why we are writing a new container runtime in Rust:
 - Rust is one of the best languages to implement the oci-runtime spec. Many very nice container tools are currently written in Go. However, the container runtime requires the use of system calls, which requires a bit of special handling when implemented in Go. This tricky (e.g. `namespaces(7)`, `fork(2)`); with Rust too, but it's not that tricky. And, unlike in C, Rust provides the benefit of memory safety. While Rust is not yet a major player in the container field, it has the potential to contribute a lot: something this project attempts to exemplify.

History: Syd-OCI

Did you exchange a walk-on part in the war for a lead role in a cage?

- Syd-OCI: The ☺ther Cønt@iner Runtime
 - Syd: Make sandboxing as easy as text searching is with `grep(1)`!
 - Watch “Syd: An Introduction to Secure Application Sandboxing for Linux”
 - Youki: `libcggroups`, `libcontainer`, `liboci-cli`
 - Syd-OCI: `s/DefaultExecutor/SydExecutor/ < youki/src/main.rs`

Overview: What?

Welcome my son, welcome to the machine.

- A secure container runtime for OCI-compliant Linux containers
- Licensed GPL-3.0, forever free
- `cargo install --features oci --locked syd`, requires `libseccomp`
- `cargo install --locked pandora_box`
- `docker run -it --runtime=syd-oci alpine`

Overview: How?

You dreamed of a big star, he played a mean guitar.

- `SYD_CONFIG_DIR`: Syd-OCI configuration directory
 - For system-wide containers: `/etc/syd/oci`
 - For rootless containers, one of the following:
 - `${XDG_CONFIG_HOME}/syd/oci` where `XDG_CONFIG_HOME` is usually `~/.config`.
 - `${HOME}/.syd/oci` if `XDG_CONFIG_HOME` is not set.
- `syd-oci` attempts to configure the `syd(1)` sandbox in the following order, and parses the first file or profile it locates and stops processing:
 - If `hostname` and `domainname` is defined for the container, try to load `${SYD_CONFIG_DIR}/${hostname}.${domainname}.syd-3`.
 - If `domainname` is defined for the container, try to load `${SYD_CONFIG_DIR}/${domainname}.syd-3`.
 - If `hostname` is defined for the container, try to load `${SYD_CONFIG_DIR}/${hostname}.syd-3`.
 - Try to load `${SYD_CONFIG_DIR}/default.syd-3`.
 - Load the builtin `oci` profile. This profile is designed to be combined with `pandora(1)` and learning mode.

Comparison: Syd-OCI vs. gVisor

Breathe, breathe in the air. Don't be afraid to care.

- Rust vs. Go!
 - Rust has idiomatic, strict error checking
 - Rust has no garbage collection, less side-effects in code
 - Allows for more refined per-syd-thread seccomp-bpf filters
 - `syd_main`, `syd_emu`, `syd_mon`, `syd_int`, ...
- PIE & ASLR, AT_SECURE, SegvGuard, {S,}ROP mitigations
- Syd-OCI is noticeably faster than gVisor.
 - Less isolation, less overhead: [sydbox.git/bench/log](https://github.com/sydbox/sydbox.git/bench/log)
 - Layered security: Landlock Houdini, CVE-2024-42318
- I NEED MORE FAST! GO FAST! UNROLL MY LOOPZ!
 - Do not panic, take your towel!
 - `trace/allow_unsafe_nice:1`
 - `trace/allow_unsafe_ptrace:1`
 - `trace/allow_unsafe_spec_exec:1`

The End

You'll lose your mind and play free games for May!

- CTF: {https,ssh}://syd.chesswob.org
 - user/pass: syd
 - rules: /etc/user.syd-3
 - goal: read /etc/CTF & get 200€!
- GIT: <https://gitlab.exherbo.org/sydbox/sydbox.git>
- DOC: <https://man.exherbolinux.org>
- ML: <https://lists.sr.ht/~alip/exherbo-dev>
- IRC: #sydbox at Libera, #sydbox:mailstation.de