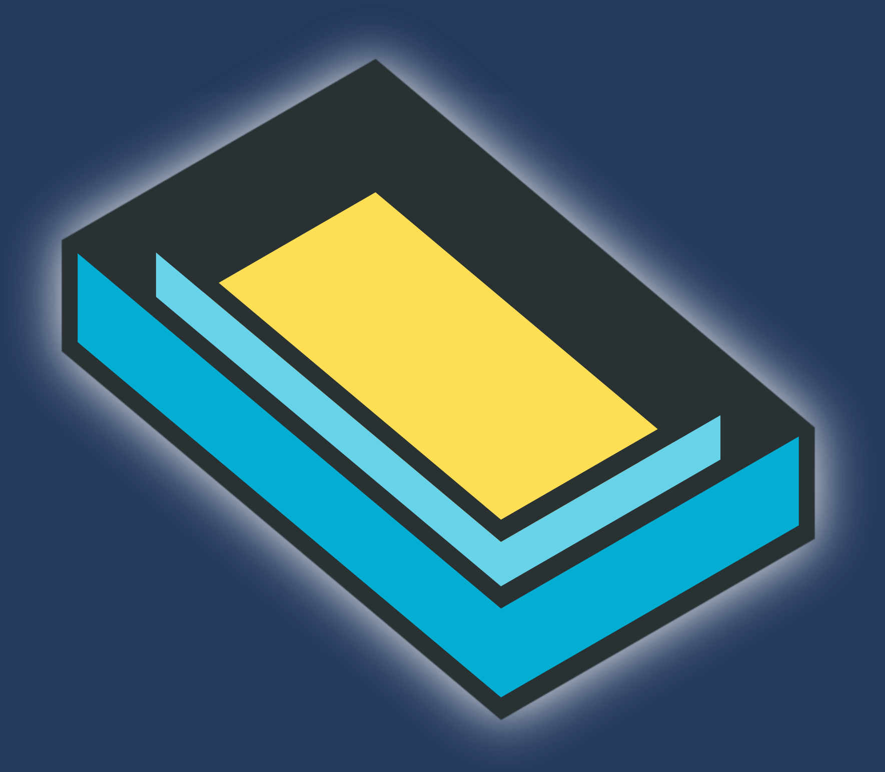
Microsoft

Sandbox IDs with Landlock

FOSDEM

Mickaël Salaün – kernel maintainer



2025-02-01

What is sandboxing?

“A **restricted**, controlled **execution environment** that prevents potentially malicious software [...] from accessing any system resources except those for which the software is authorized.”

Landlock

- Access control system (orthogonal to namespaces)
- Dynamic security policies
- Embeddable in apps/services: unprivileged
- Enabled by default on most distros

Various use cases for IDs

Containers, IMA/EVM, audit…

Container label/ID properties

- Inherited from process to process
- ~~Immutable~~ only extendable (e.g., strings)
- Global for privileged services
- Relative for unprivileged services
- Persistence uniqueness for attestation (e.g., 128-bit UUID)
- Predictable ID for attestation?
- CRIU support

Landlock properties

Use case #1

Untrusted applications: protect from potentially malicious third-party code.

Candidates:

- Container runtimes
- Init systems

Use case #2

Exploitable bugs in trusted applications: protect from vulnerable code maintained by developers.

Candidates:

- Parsers: archive tools, file format conversion, renderers…
- Web browsers
- Network and system services

Current access control

Implicit restrictions

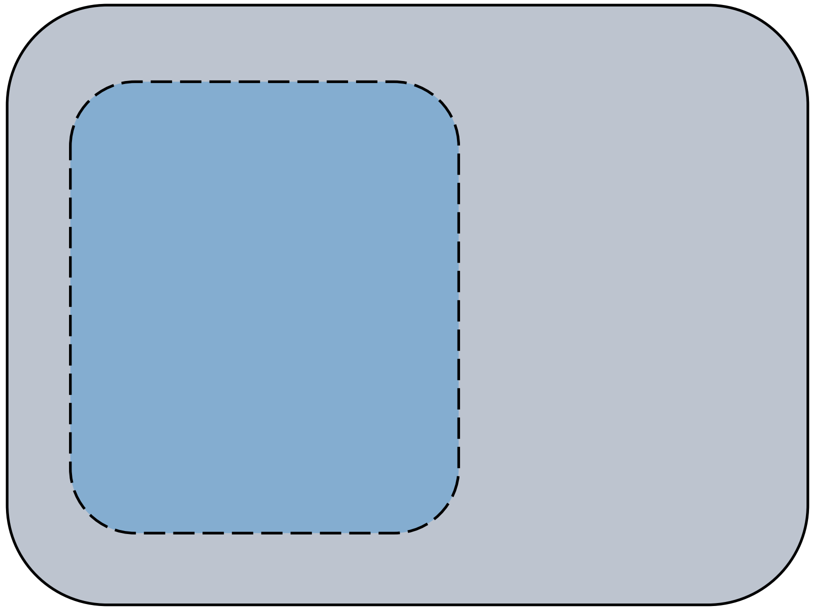
- Process impersonation (e.g., ptrace)
- Filesystem topology changes (e.g., mounts), when it makes sense

Explicit access rights

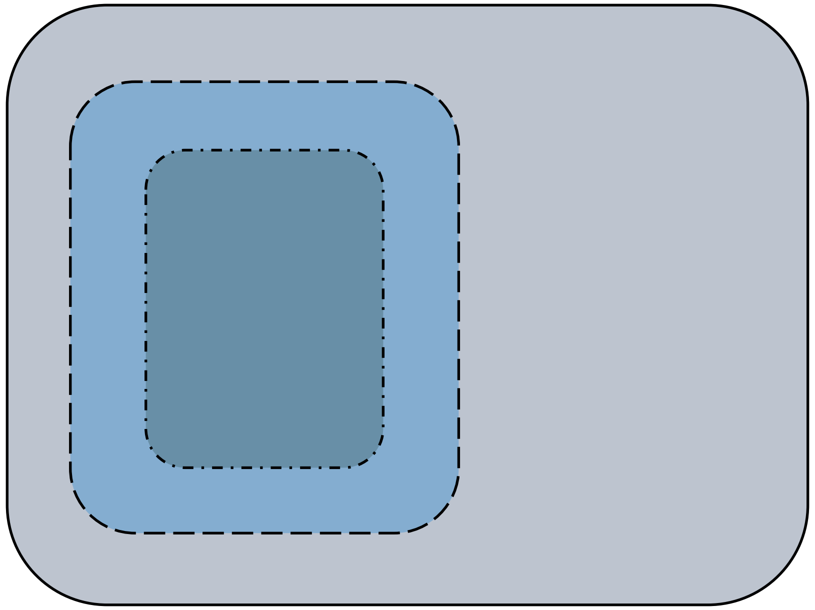
- Filesystem
- Networking
- Signaling
- Abstract unix socket

Dynamic policy composition

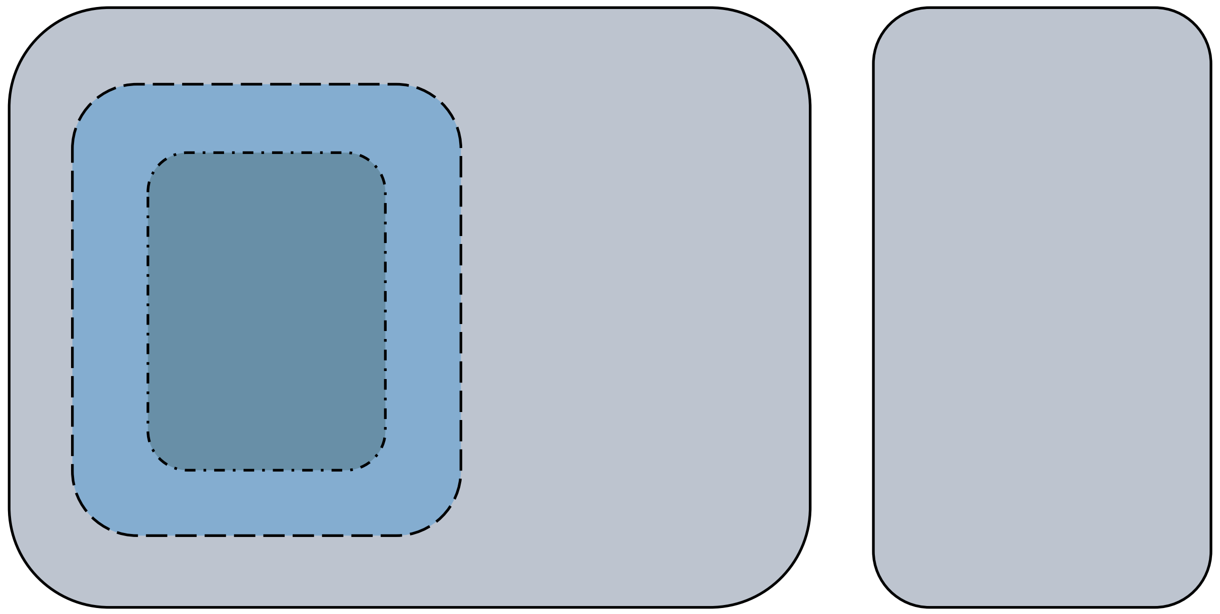
Dynamic policy composition



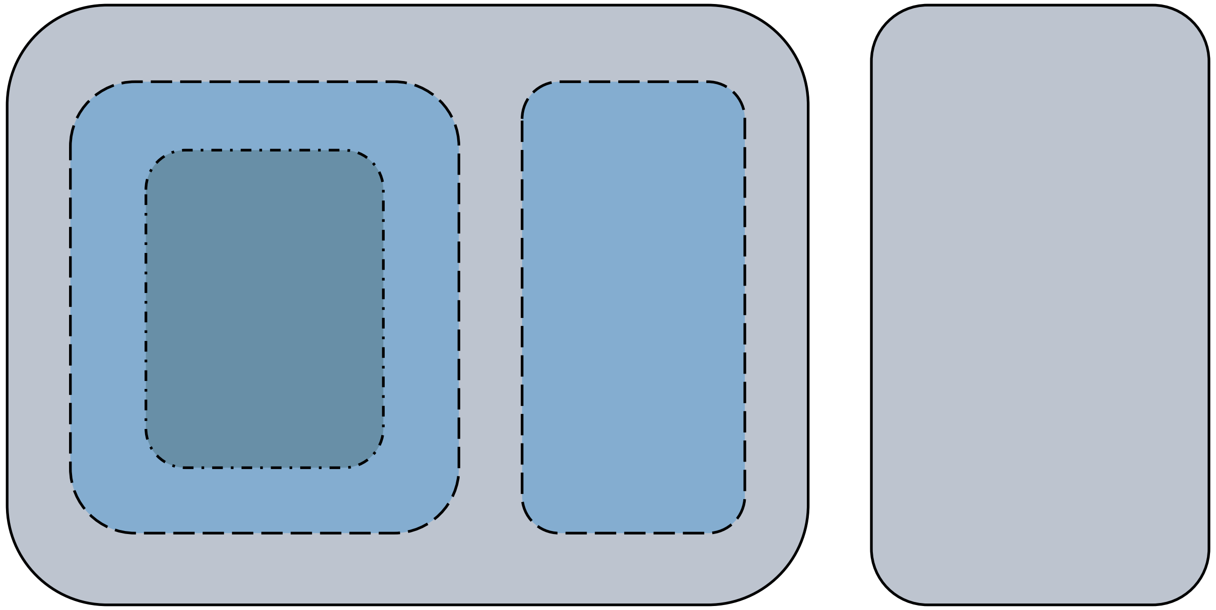
Dynamic policy composition



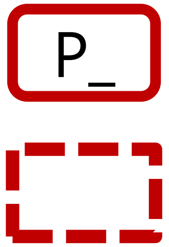
Dynamic policy composition



Dynamic policy composition



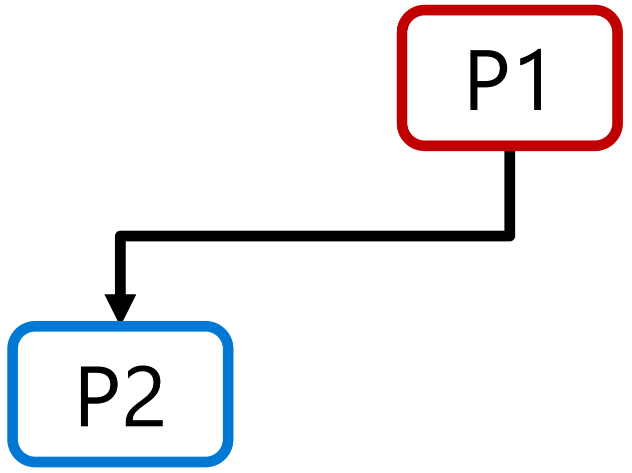
Inherited security policies

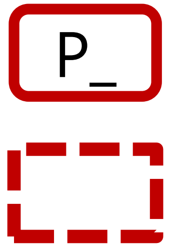


Sandboxed process

Sandbox domain

Inherited security policies

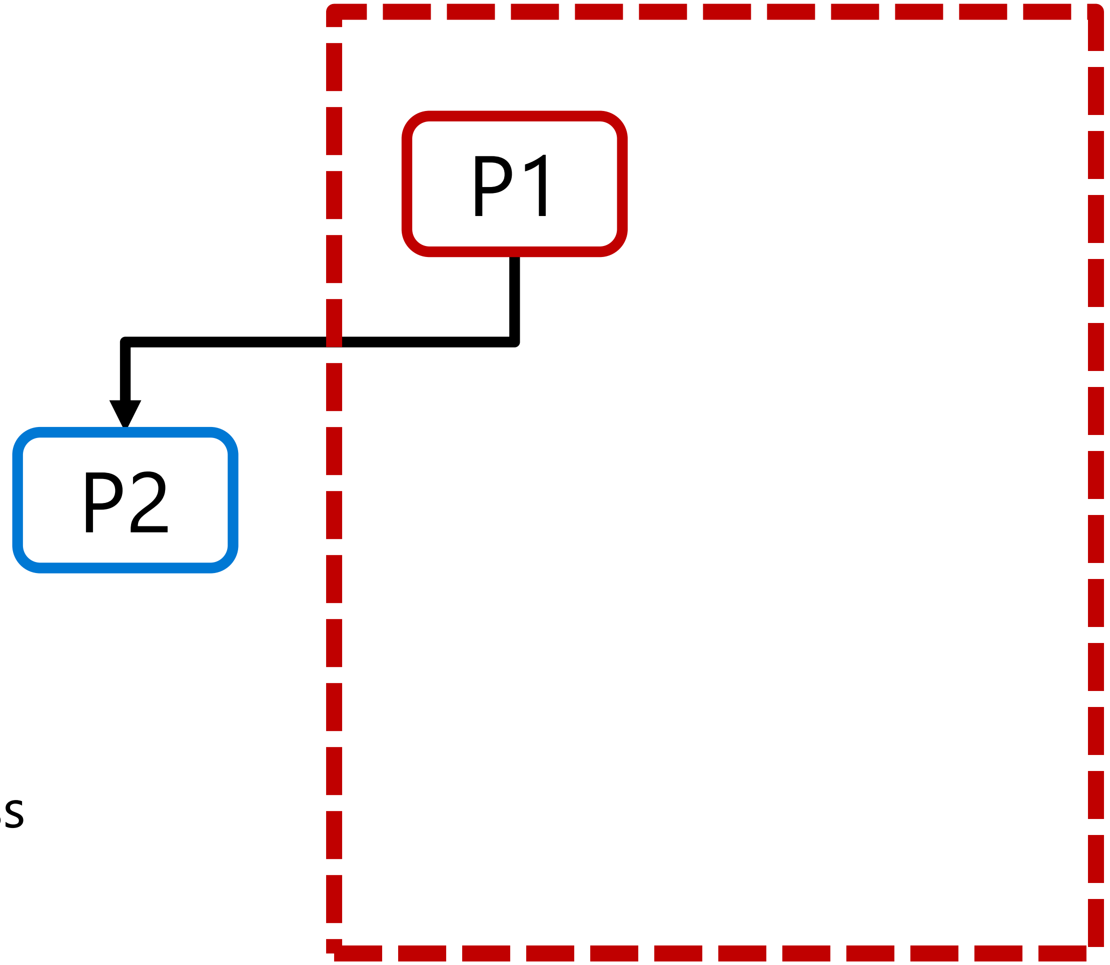


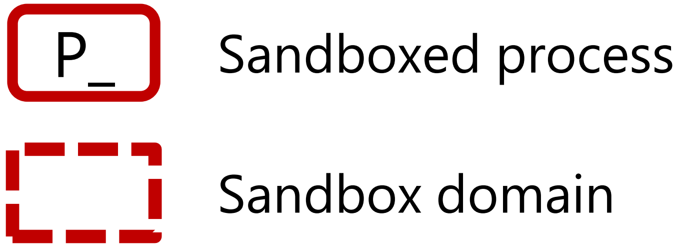


Sandboxed process

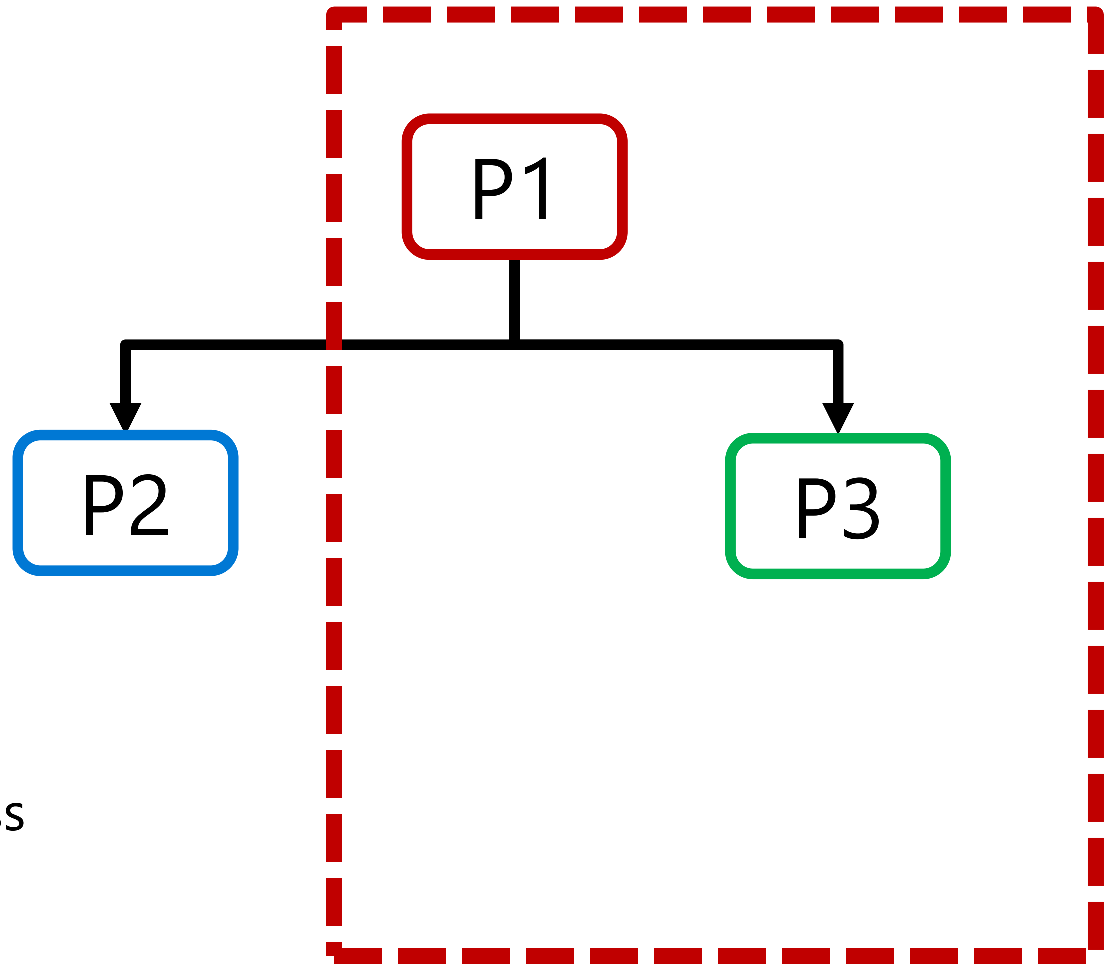
Sandbox domain

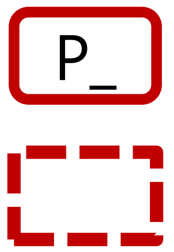
Inherited security policies





Inherited security policies

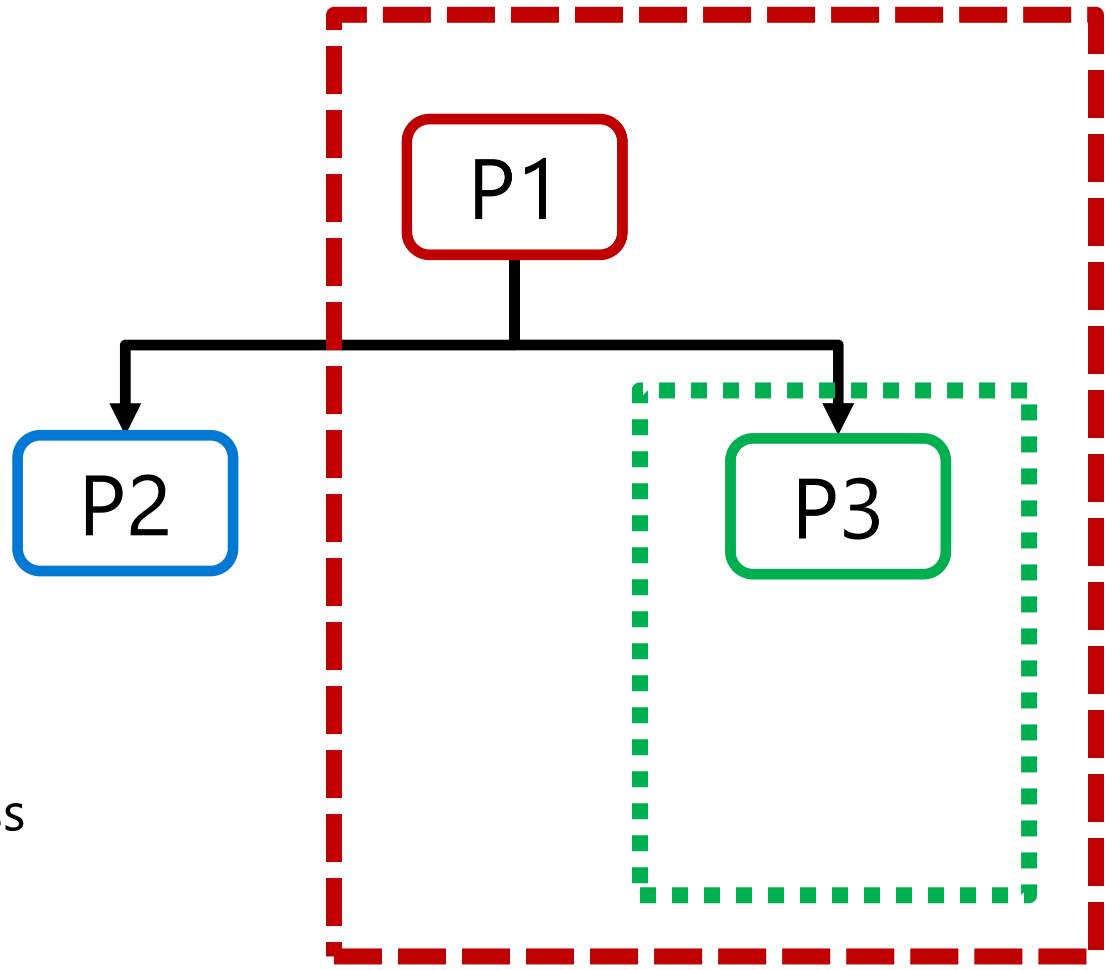


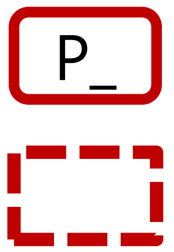


Sandboxed process

Sandbox domain

Inherited security policies

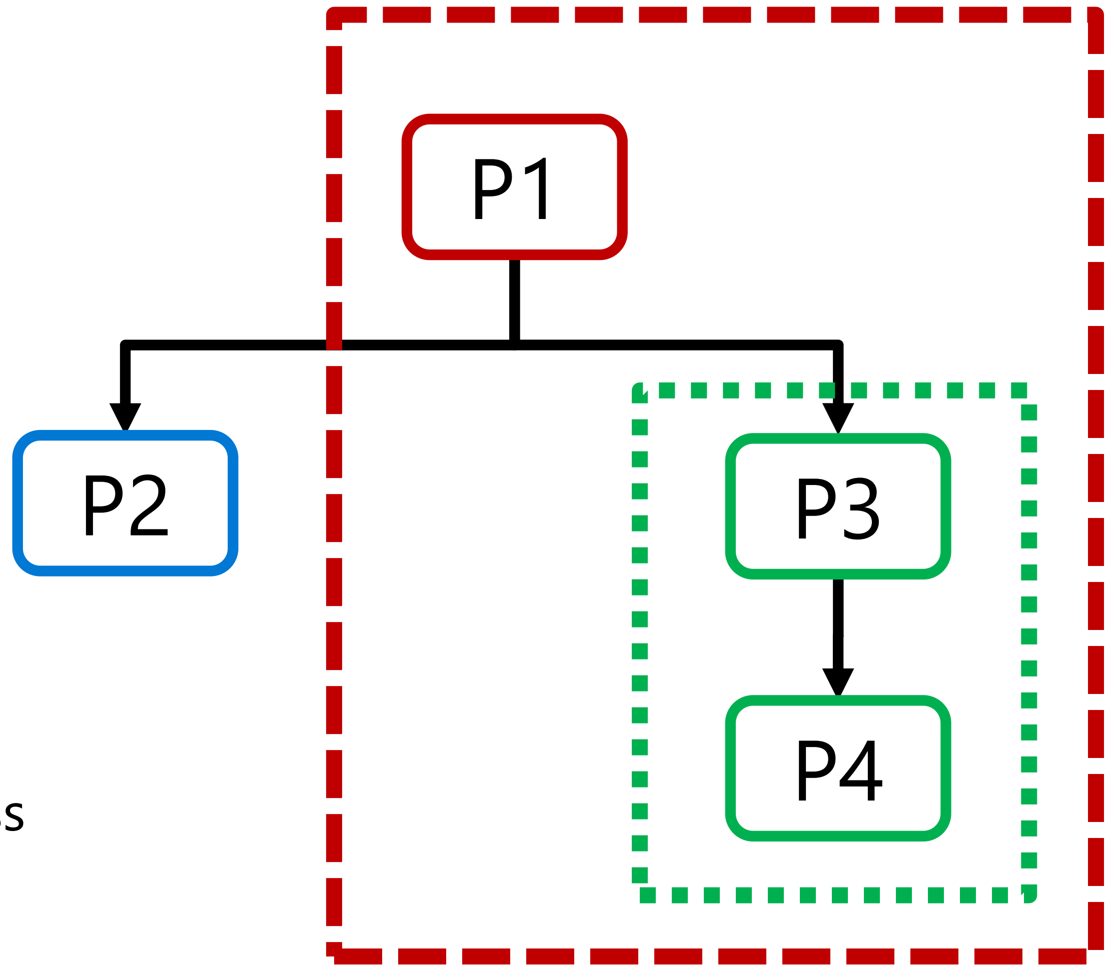


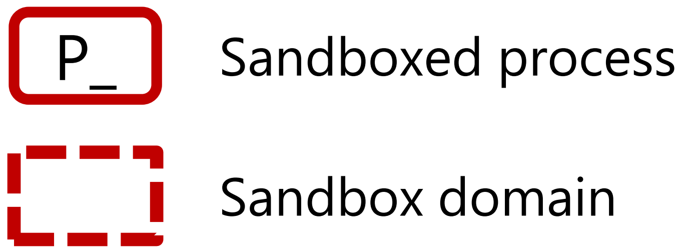


Sandboxed process

Sandbox domain

Inherited security policies





Landlock domain IDs

Landlock domain ID properties

- Unique during the lifetime of the running system: ~2^{60} IDs
- First value randomly picked between 2^{32} and 2^{33}
  - Force u64 type to limit parsing issues
  - Limit collision in logs, can be concatenated with the boot ID for fleet-unique IDs
  - Mostly aligned IDs: ~same numbers of (hexadecimal) characters
- Sequential but not necessarily consecutives IDs: incremented between 1 and 16
  - Limit cover channels
  - Still expose sequentially domain creation: useful for domain ordering and to optimize ID lookup

Use cases for Landlock IDs

Because of nested sandboxes there are two main use cases:

1. You create a sandbox and want to identify if a process is in this sandbox
2. You want to identify the latest layer of sandboxing restricting a task (i.e., the full sandbox)

pidfd

- File descriptor referencing a process:
  - Proper kernel object with clear lifetime
  - Avoid race conditions (e.g., TOCTOU)
- Created from a PID or from a unix socket to identify a peer
- Used to send signal, wait… and read process properties thanks to the new PIDFD_GET_INFO IOCTL (for a set of properties):
  - PIDFD_INFO_CREDS
  - PIDFD_INFO_CGROUPID…

Extended PIDFD_GET_INFO

Two new PIDFD_GET_INFO flags:

- PIDFD_INFO_LANDLOCK_LAST_DOMAIN
- PIDFD_INFO_LANDLOCK_FIRST_DOMAIN

[RFC PATCH v1 0/3] Expose Landlock domain IDs via pidfd

Future work

- Add a new interface for CRIU
- Add a dedicated introspection interface to safely read all properties of a sandbox; some ideas:
  - Properties (e.g., “comm”) of the process that sandbox itself, which could be used to give a name to sandboxes, or to **label containers**?
  - Read properties of Landlock rulesets used to create domains
  - Walk through domain hierarchies
  - Get notifications about denied access requests…

Could these IDs be used for other use cases?

- ✓ Inherited from process to process
- ✓ Immutable and extendable (e.g., ~~strings~~)
- ✓ Global for privileged services
- ✓ Relative for unprivileged services
- ✓ Persistence uniqueness for attestation: add boot ID to a Landlock domain ID? (e.g., 128-bit UUID)
- ✗ Predictable ID for attestation?
- ✓ Predictable label for attestation?
- ! CRIU support

Wrap-up

Landlock roadmap

Ongoing work:

- Audit support to ease debugging and provide metrics
- Introspection interfaces (e.g., pidfd)
- New access-control types: socket creation, UDP port use…

Contribute

- Develop new access types or features
- Improve libraries: Rust, Go…
- Challenge the implementation
- Improve documentation or tests
- Sandbox your programs and others’
  - Secure Open Source Rewards
  - Google Patch Rewards

Try Landlock

```
# WARNING: The "sandboxer" is a demonstration program,
# not a tool with a stable interface.

$ cargo install landlock --examples

$ sandboxer
```

Questions?

landlock@lists.linux.dev

Thank you!