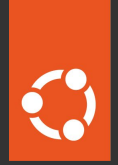


Restricting Unprivileged User Namespaces In Ubuntu

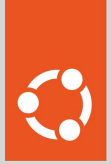
John Johansen, Maxime Bélair
September 17, 2024

User Namespaces

- Isolate security attributes
  - User/group ID, root directory, capabilities, …
  - Different namespaces can have identical user and group ID ranges without conflict.

- A process can be
  - Root privileges inside a namespace
  - Regular user on the host system.

- Base brick for containerization.



Problem

Unprivileged user namespaces
Allow unprivileged users to access kernel vulnerabilities
that would normally be root only

Is it really a problem?

CVE-2024-1086: to exploit, needs to be able to add netfilter rules, granted by `CAP_NET_ADMIN` in a new user and network namespace.

CVE-2023-35001: to exploit, needs to be able to add nftables rules, granted by `CAP_NET_ADMIN` in a new user and network namespace.

CVE-2022-32250: to exploit, needs to be able to add netfilter rules, granted by `CAP_NET_ADMIN` in a new user and network namespace.

CVE-2022-0185: to exploit, need to be able to mount a filesystem, granted by `CAP_SYS_ADMIN` in a user namespace.

CVE-2022-1015: to exploit, need to be able to add netfilter rules, granted by `CAP_NET_ADMIN` in a new user and network namespace.

CVE-2022-2078: to exploit, need to be able to add netfilter rules, granted by `CAP_NET_ADMIN` in a new user and network namespace.

CVE-2022-24122: reference counting error when leaving a user namespace.

CVE-2022-25636: to exploit, need to be able to add netfilter rules, granted by `CAP_NET_ADMIN` in a new user and network namespace.

CVE-2020-14386: to exploit, need to interact with `AF_PACKET`, granted by `CAP_NET_RAW` in a new user namespace.

CVE-2020-16120: to exploit, needs to be able to mount fuse overlay and shiftfs.

…

Ubuntu pwn2own 2017, 2020, 2021, 2022, 2023, 2024

Syzkaller reporting ~1000 bugs a year

Google report 44% of the exploits they saw required unprivileged user namespaces
https://security.googleblog.com/2023/06/learnings-from-kctf-vrps-42-linux.html

Goal

Prevent
Untrusted Unprivileged users/code
from using
unprivileged user namespaces
to attack kernel

Goal - Restated

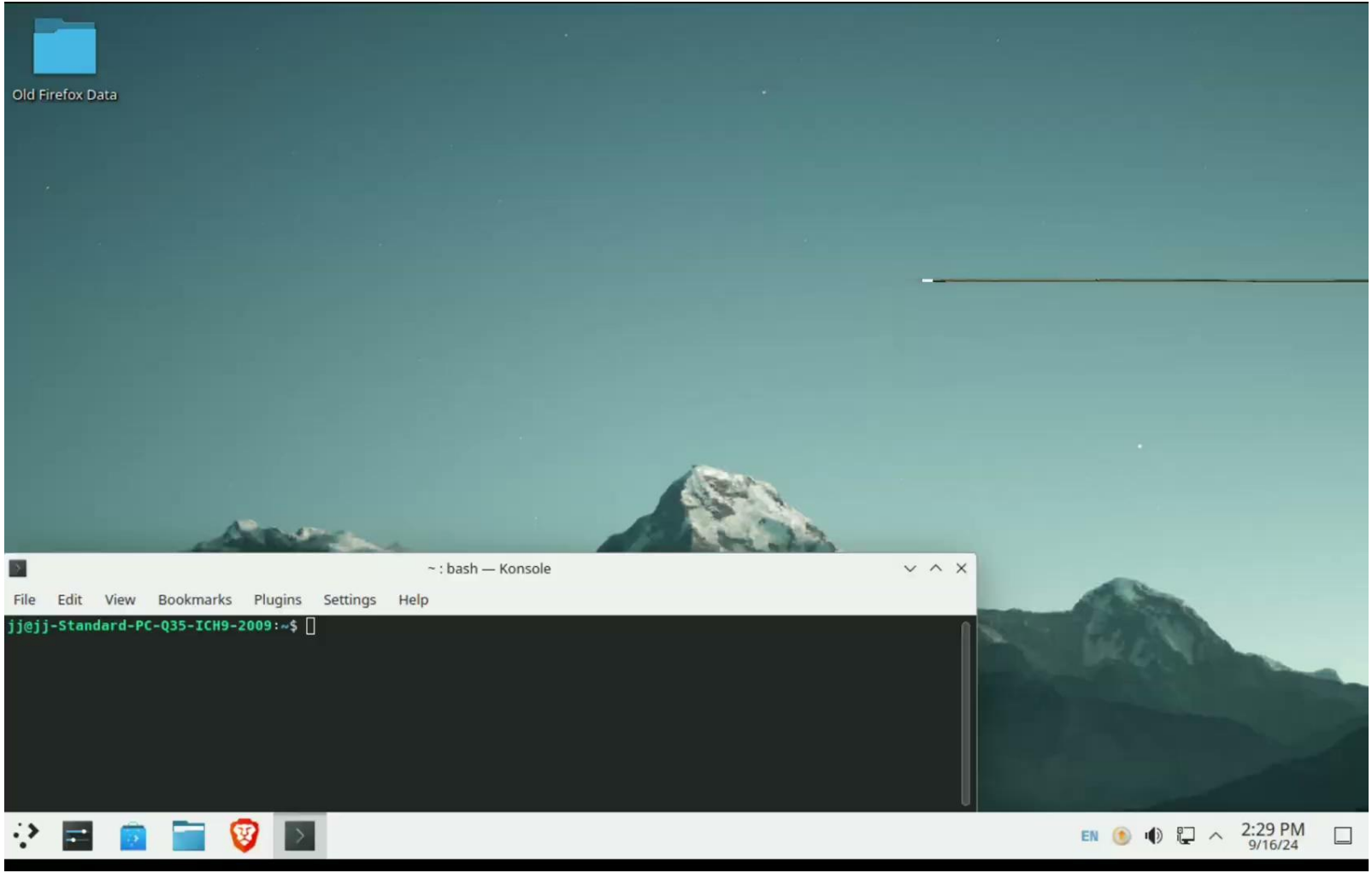
Make
unprivileged user namespaces
Privileged
without breaking the world

Solution #1

Restrict User Namespace Creation

- Enable by default in kernel

- Restrict userns creation to only trusted applications
  - policy allows “trusted” applications to create
- “unconfined”
  - Unprivileged user can not create
  - Privileged user can
- Global policy variable
  - Make it easy for the admin to disable


Old Firefox Data
~ : bash — Konsole
File Edit View Bookmarks Plugins Settings Help
jj@jj-Standard-PC-Q35-ICH9-2009:~$
EN
2:29 PM
9/16/24

AppArmor user namespace creation restrictions cause many applications to crash with SIGTRAP

Bug #2046844 reported by Xavier Guillot on 2023-12-18

This bug affects 44 people. Does this bug affect you?

288

Affects	Status	Importance	Assigned to	Milestone
AppArmor	New	Undecided	Unassigned	
Wike	New	Unknown	auto-github-hugolabe-wike #181	
akonadiconsole (Ubuntu)	Fix Released	High	Scarlett Gately Moore	Ubuntu ubuntu-24.04-feature-freeze
akregator (Ubuntu)	Fix Released	Critical	Scarlett Gately Moore	Ubuntu ubuntu-24.04-feature-freeze
angelfish (Ubuntu)	Fix Released	Critical	Scarlett Gately Moore	Ubuntu ubuntu-24.04-feature-freeze
apparmor (Ubuntu)	Fix Released	Critical	Unassigned	Target to milestone
Noble	Fix Committed	Undecided	Unassigned	Target to milestone
bubblewrap (Ubuntu)	Fix Committed	Critical	Unassigned	Target to milestone
cantor (Ubuntu)	Fix Released	Critical	Scarlett Gately Moore	Ubuntu ubuntu-24.04-feature-freeze
devhelp (Ubuntu)	Fix Released	Undecided	Georgia Garcia	Target to milestone
digikam (Ubuntu)	Fix Released	High	Scarlett Gately Moore	Ubuntu ubuntu-24.04-feature-freeze
epiphany-browser (Ubuntu)	Fix Released	High	Georgia Garcia	Target to milestone

Report a bug

This report contains **Public** information

Everyone can see this information.

- Mark as duplicate
- Convert to a question
- Link a related branch
- Link to CVE
- Change lock status

Duplicates of this bug

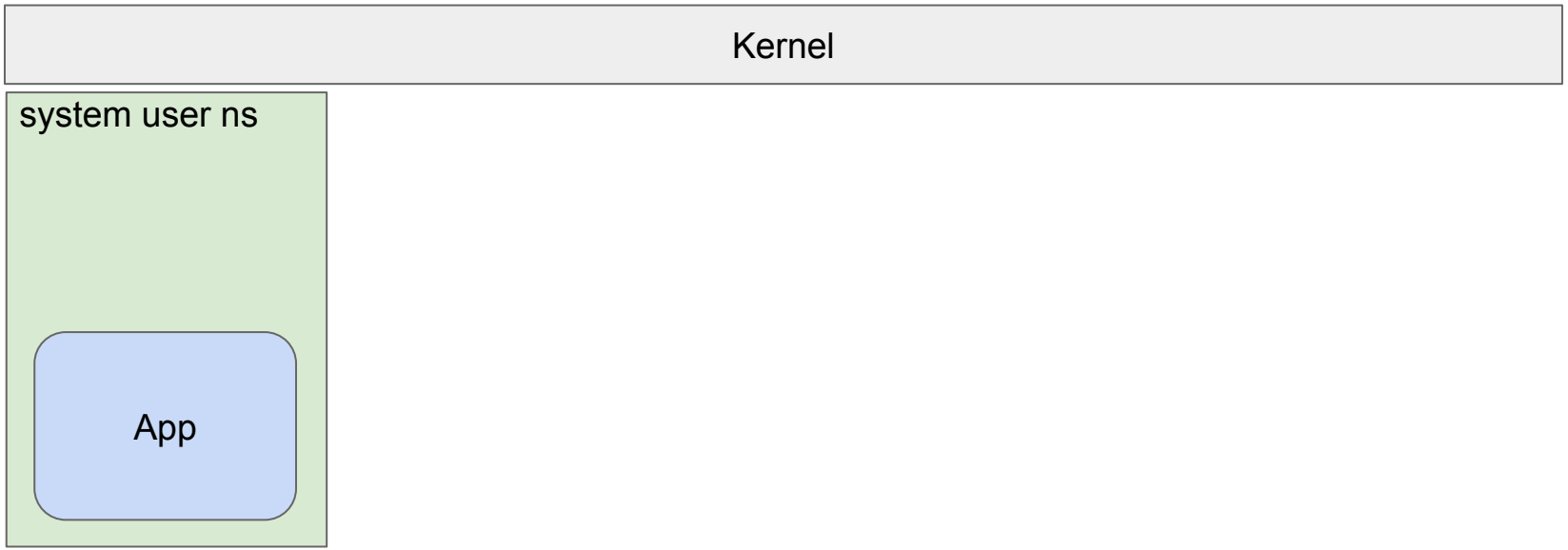
- Bug #2046624
- Bug #2046796
- Bug #2046801
- Bug #2047282
- Bug #2047986
- Bug #2049240
- Bug #2052491
- Bug #2054142
- Bug #2055725
- Bug #2055973
- Bug #2056123
- Bug #2056190
- Bug #2056297
- Bug #2064781

You have subscriptions that may cause you to receive notifications, but you are not directly subscribed to this bug's notifications.

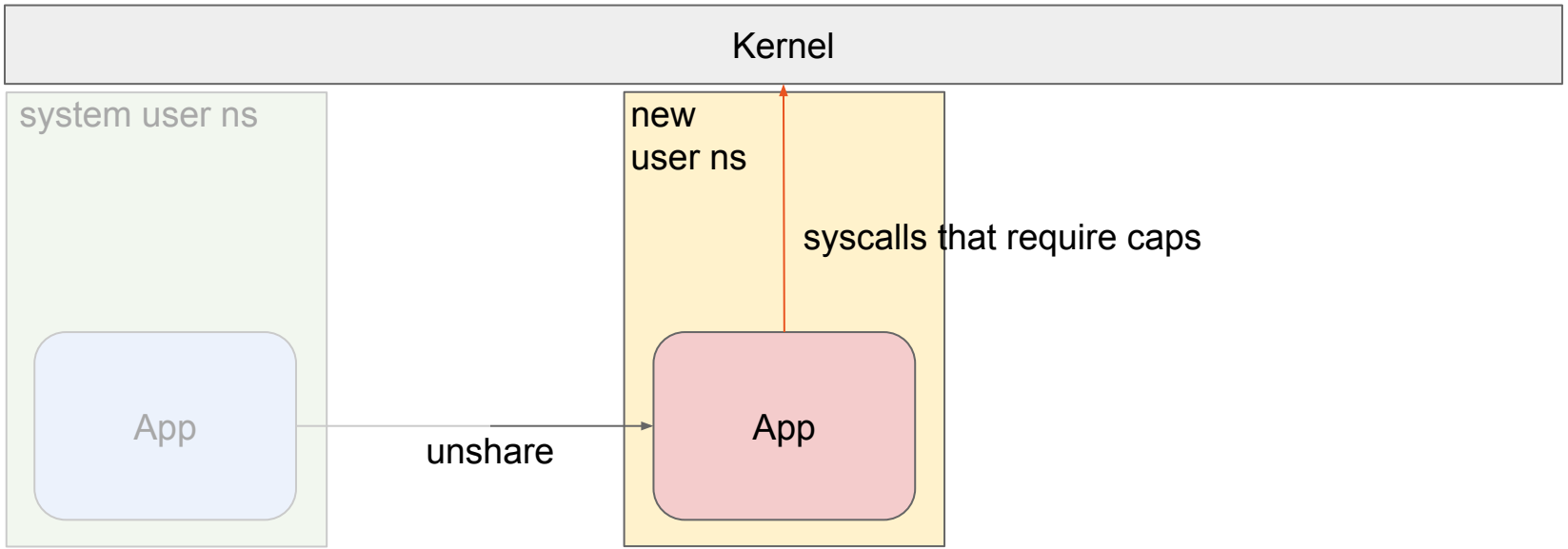
Mute bug mail

How are user namespaces used?

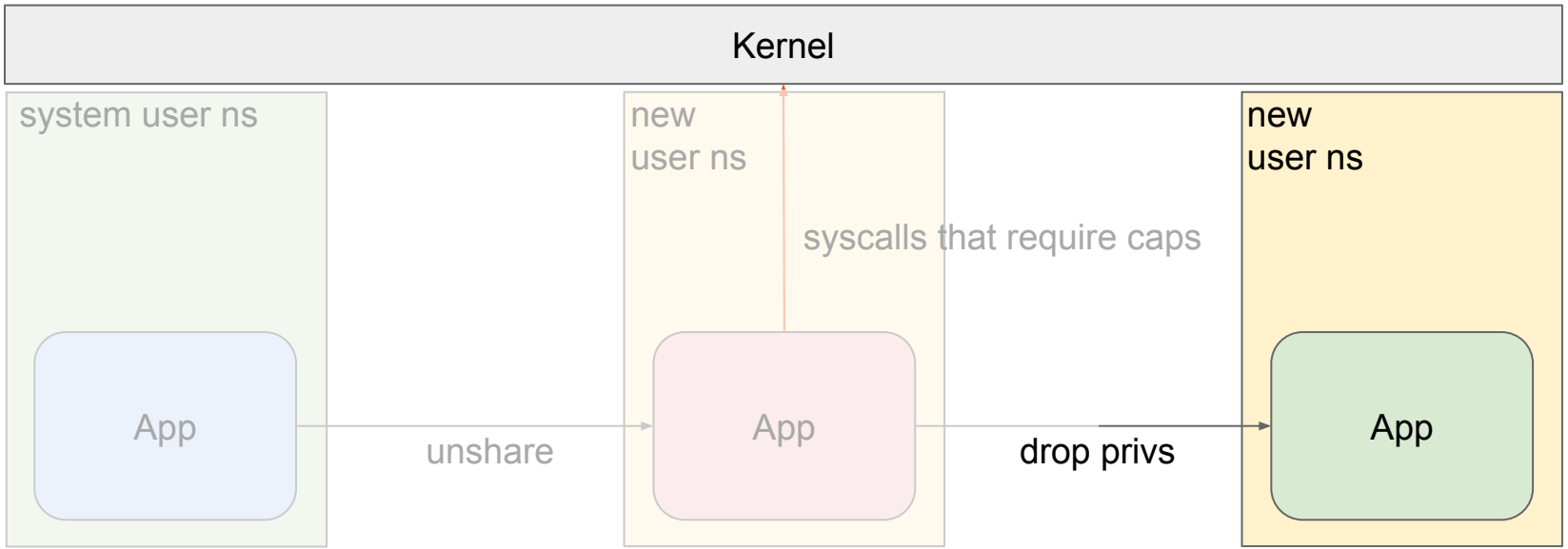
Basic use of unshare



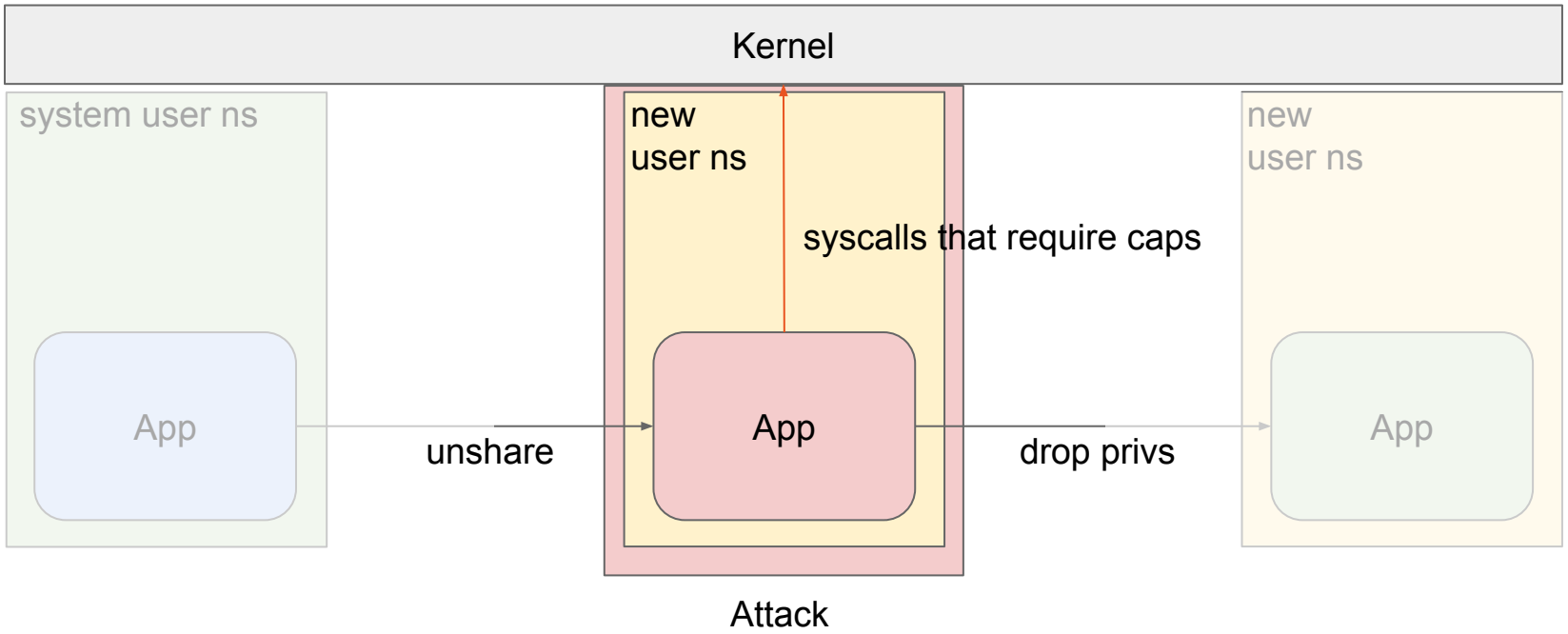
Basic use of unshare - setup



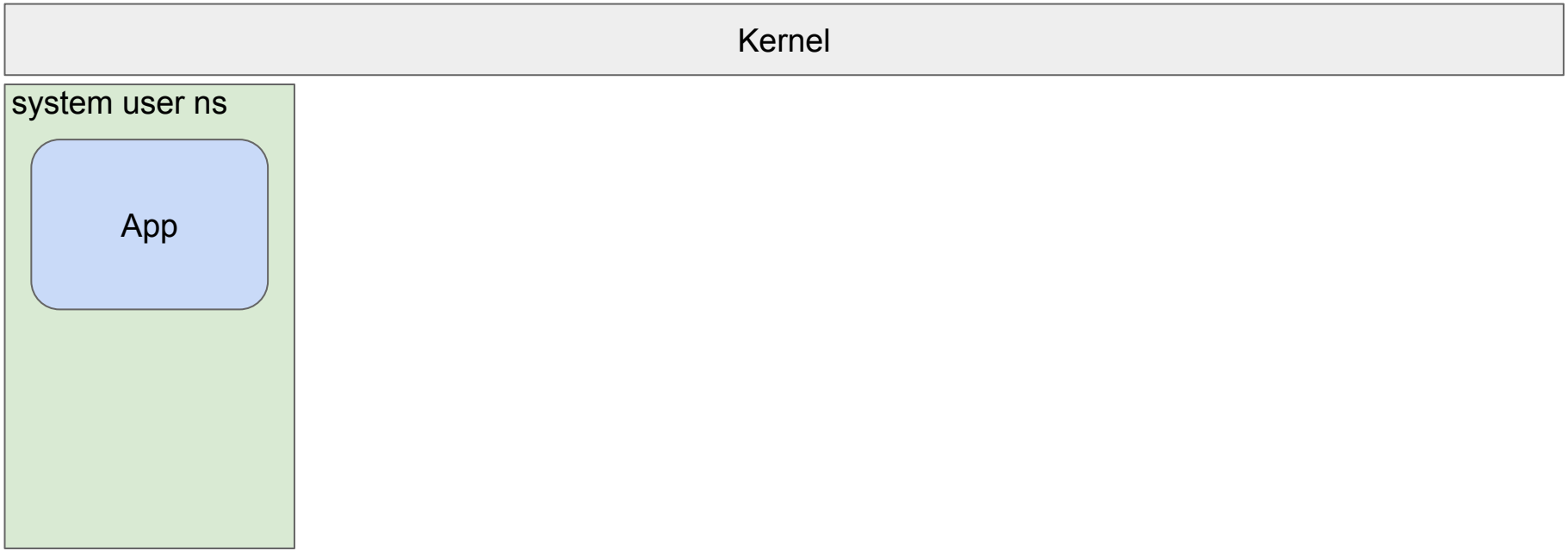
Basic use of unshare - sandbox



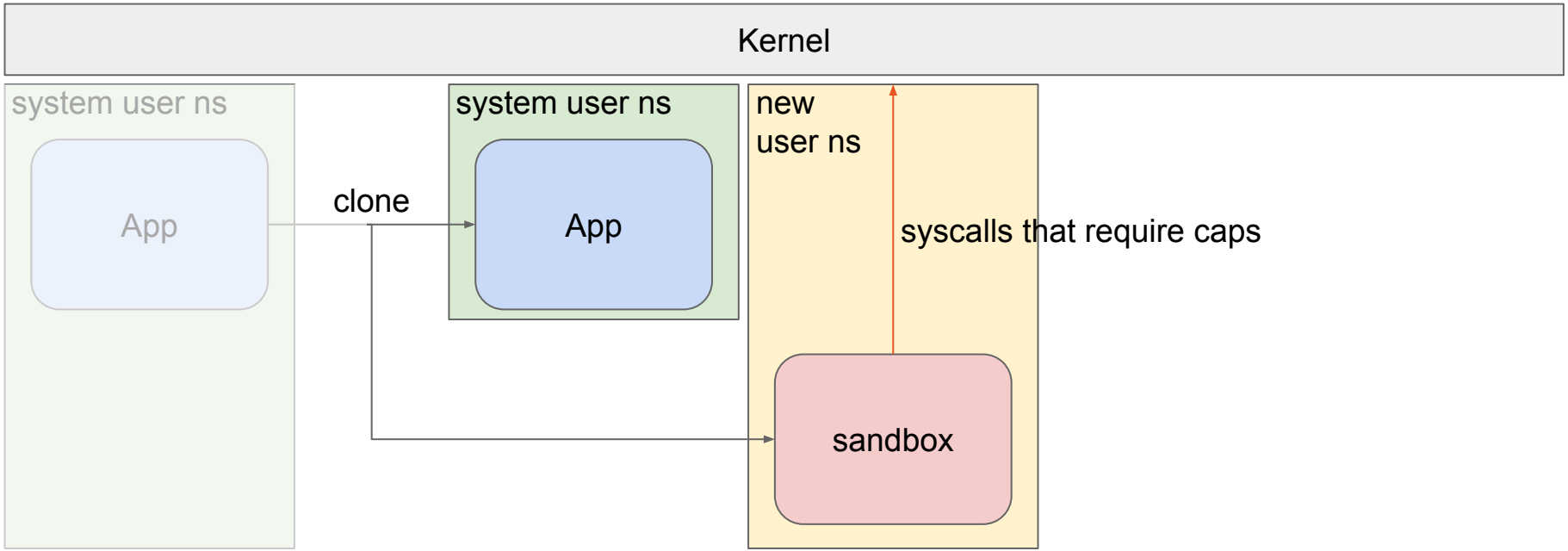
Basic use of unshare - attack



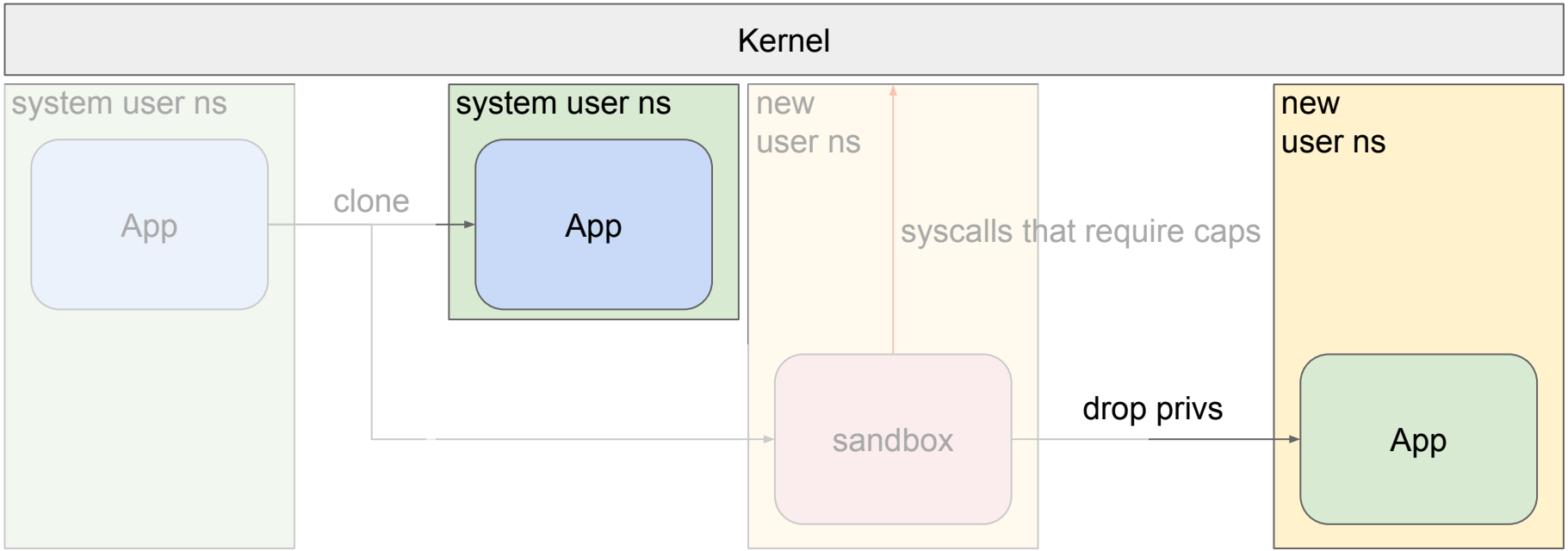
Basic use of clone



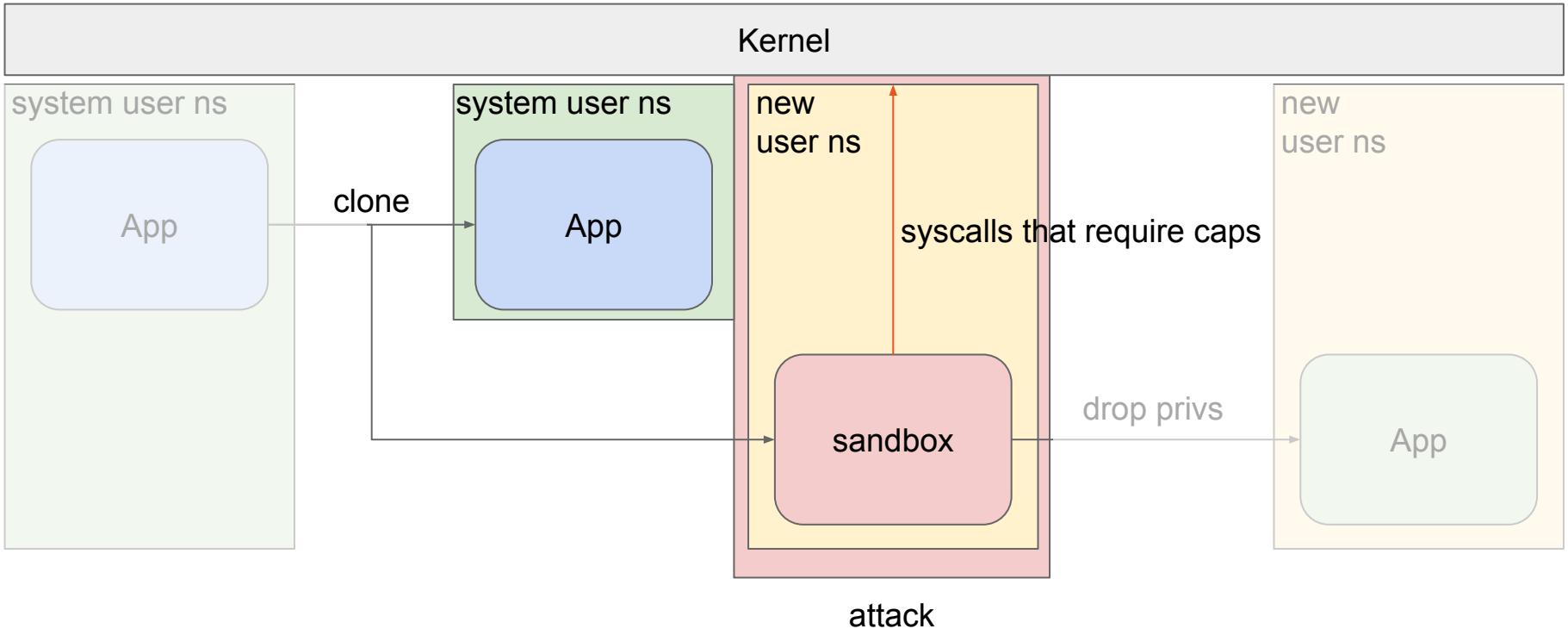
Basic use of clone - setup



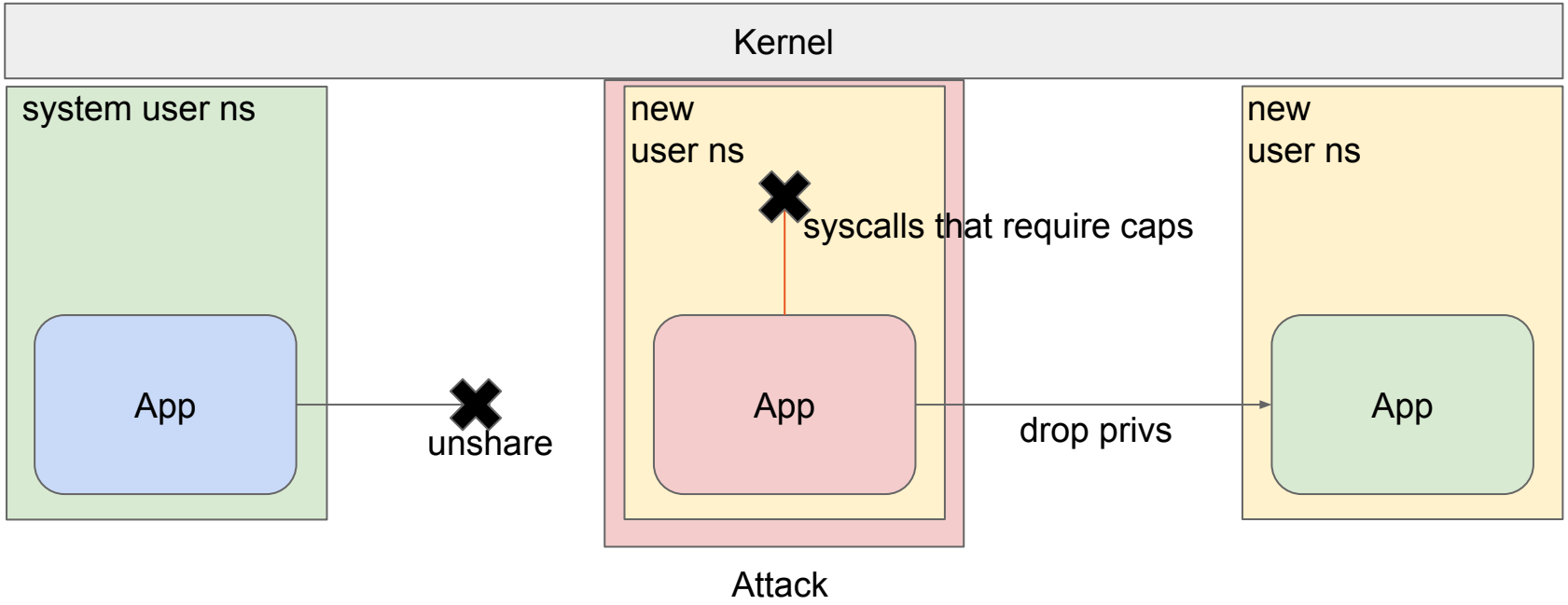
Basic use of clone - sandbox



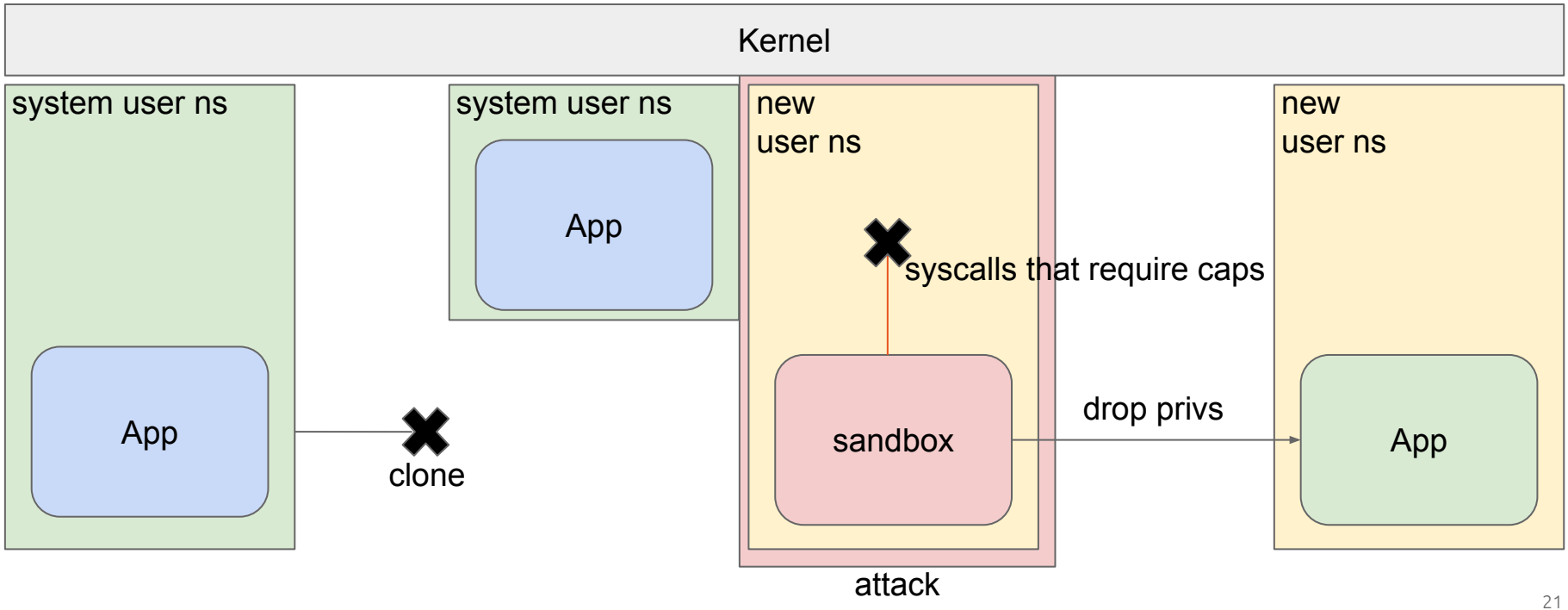
Basic use of clone - attack



Basic use of unshare - blocking attack



Basic use of clone - blocking attack



Two options to block the threat

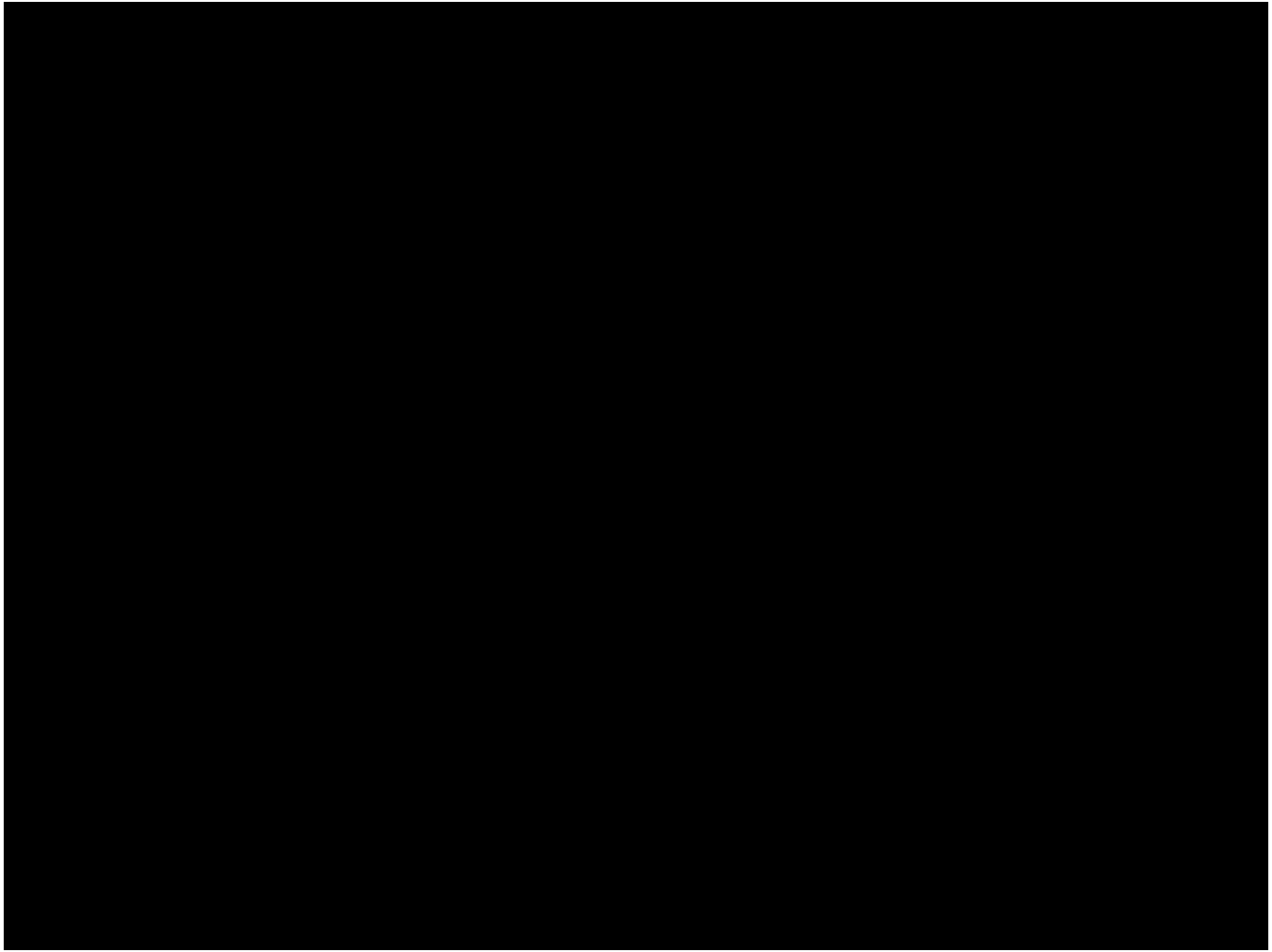
- Block userns creation
  - Straight forward
  - Easy to attribute crashes

```
apparmor="DENIED" operation="userns_create" profile="firefox"
```

- Block capabilities
  - Indirect
    - Not always clear that the capability requests are due to the userns

```
apparmor="DENIED" operation="capable" profile="firefox" capname="net_admin"
```

      - Need to know to look for userns
  - Can be selective
    - May require more MAC rules than just for the capability



Applications Error Handling

Deny userns creation

vs.

Denying Capabilities

Solution #2

Restrict User Namespace Creation

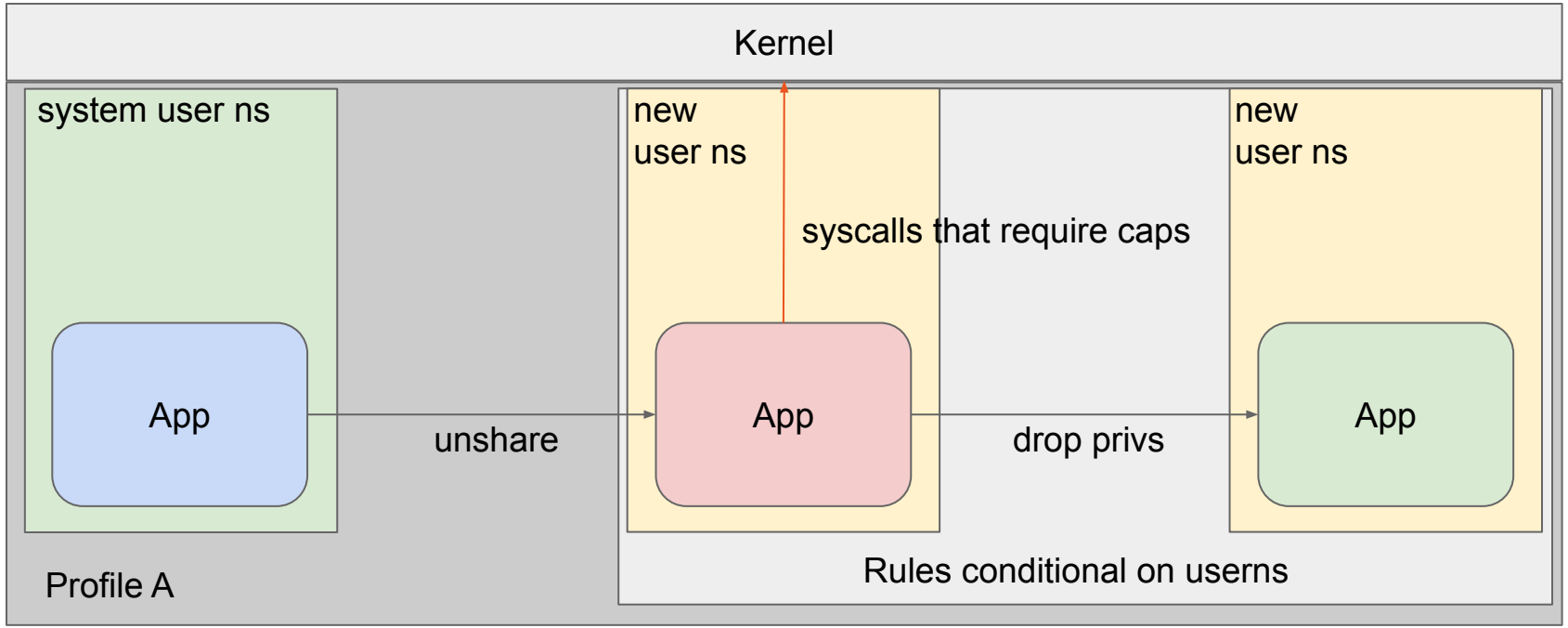
- Enable by policy on boot

- Restrict userns creation to only trusted applications
  - policy mediates
    - userns creation
    - mediate which capabilities are allowed
- “unconfined”
  - userns transitions to **unprivilege_userns** profile
  - **unprivileged_userns** denies capabilities
- Global policy variable
  - Make it easy for the admin to disable

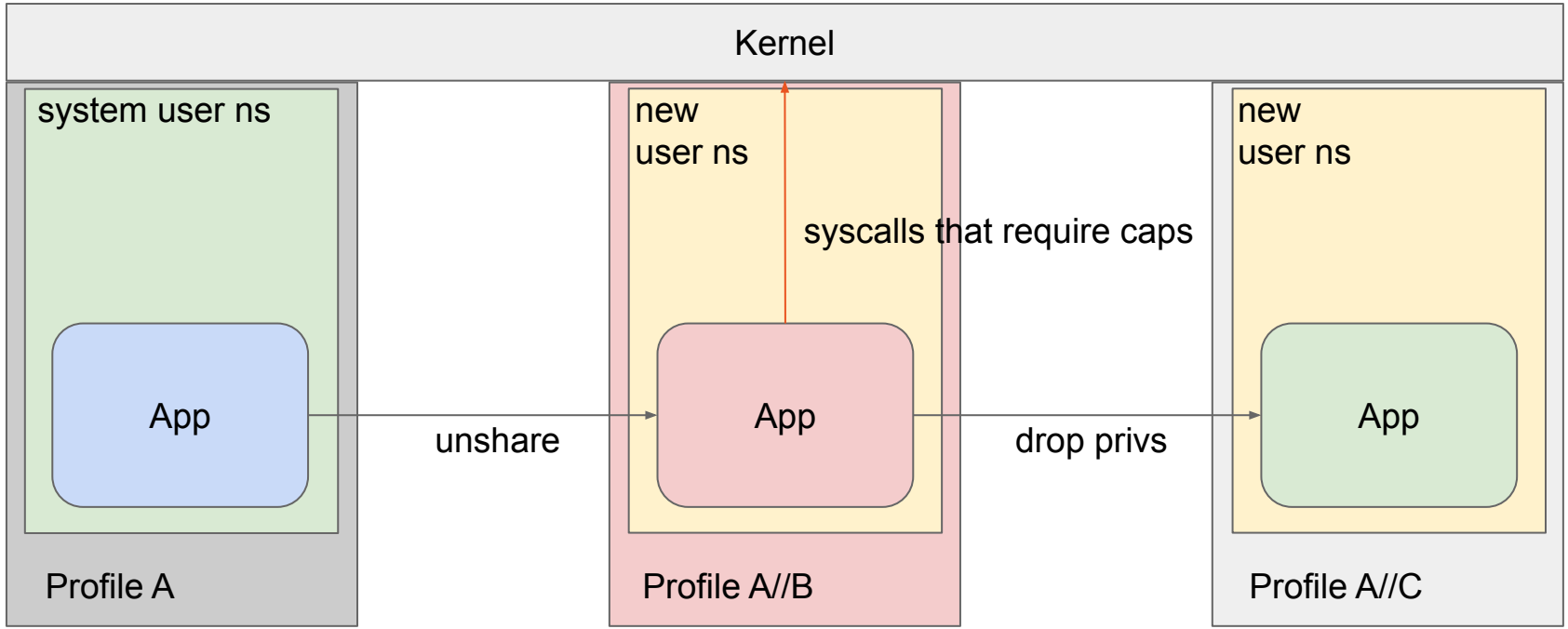
unprivileged_userns

```
abi <abi/4.0>,
include <tunables/global>

profile unprivileged_userns {
     audit deny capability,
     audit deny change_profile,

     allow network,
     allow signal,
     allow dbus,
     allow file rwlkm /**,
     allow userns,
     # ...

     allow pix /** -> &unprivileged_userns ,
}
```

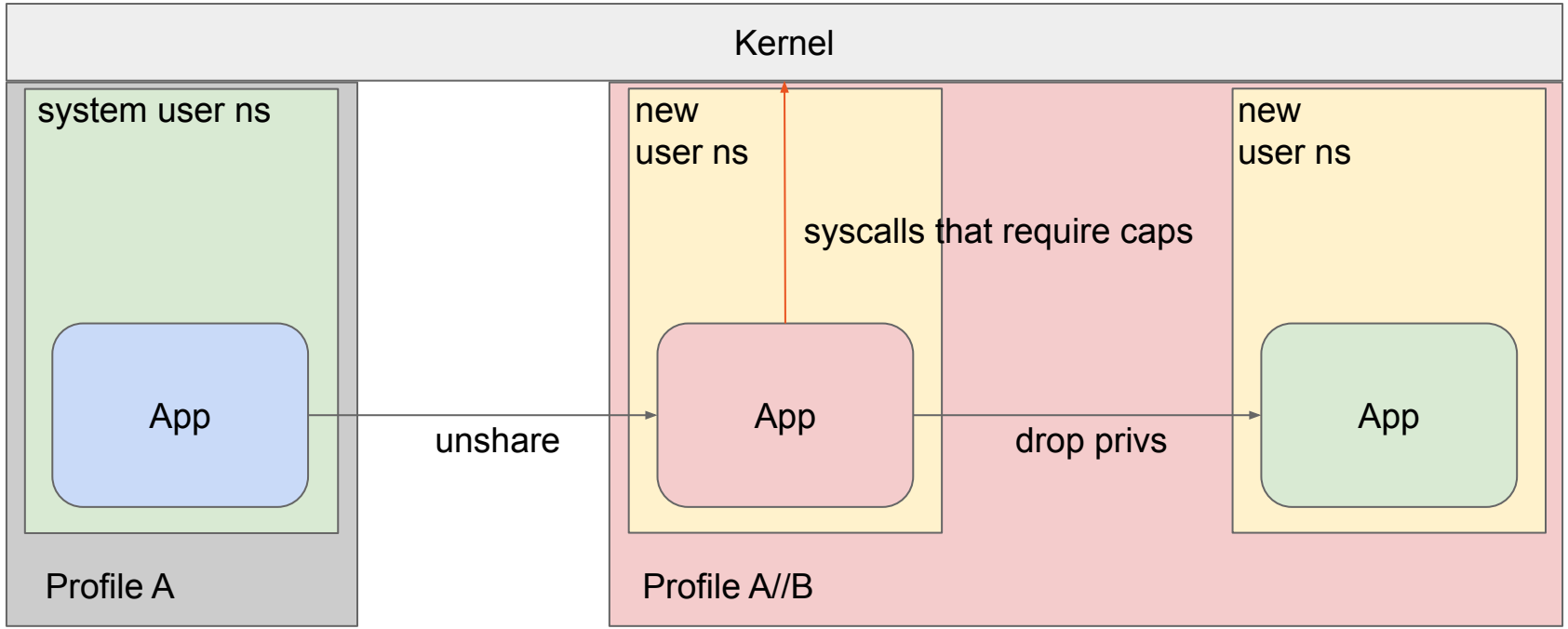
Profile layout - rules conditional on userns



Profile layout - Ideal



Profile layout - Not Ideal



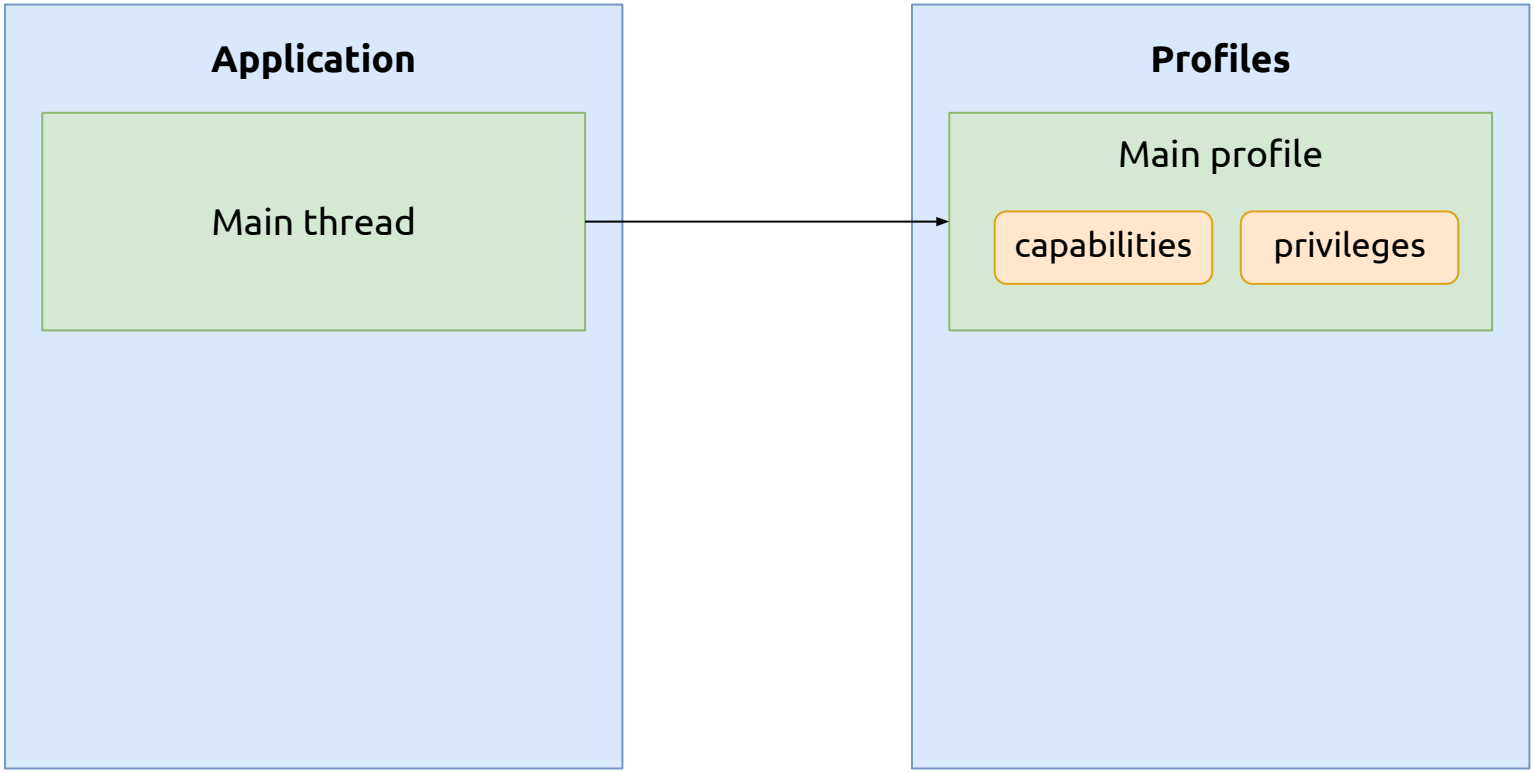
Conditional in userns vs. Transition

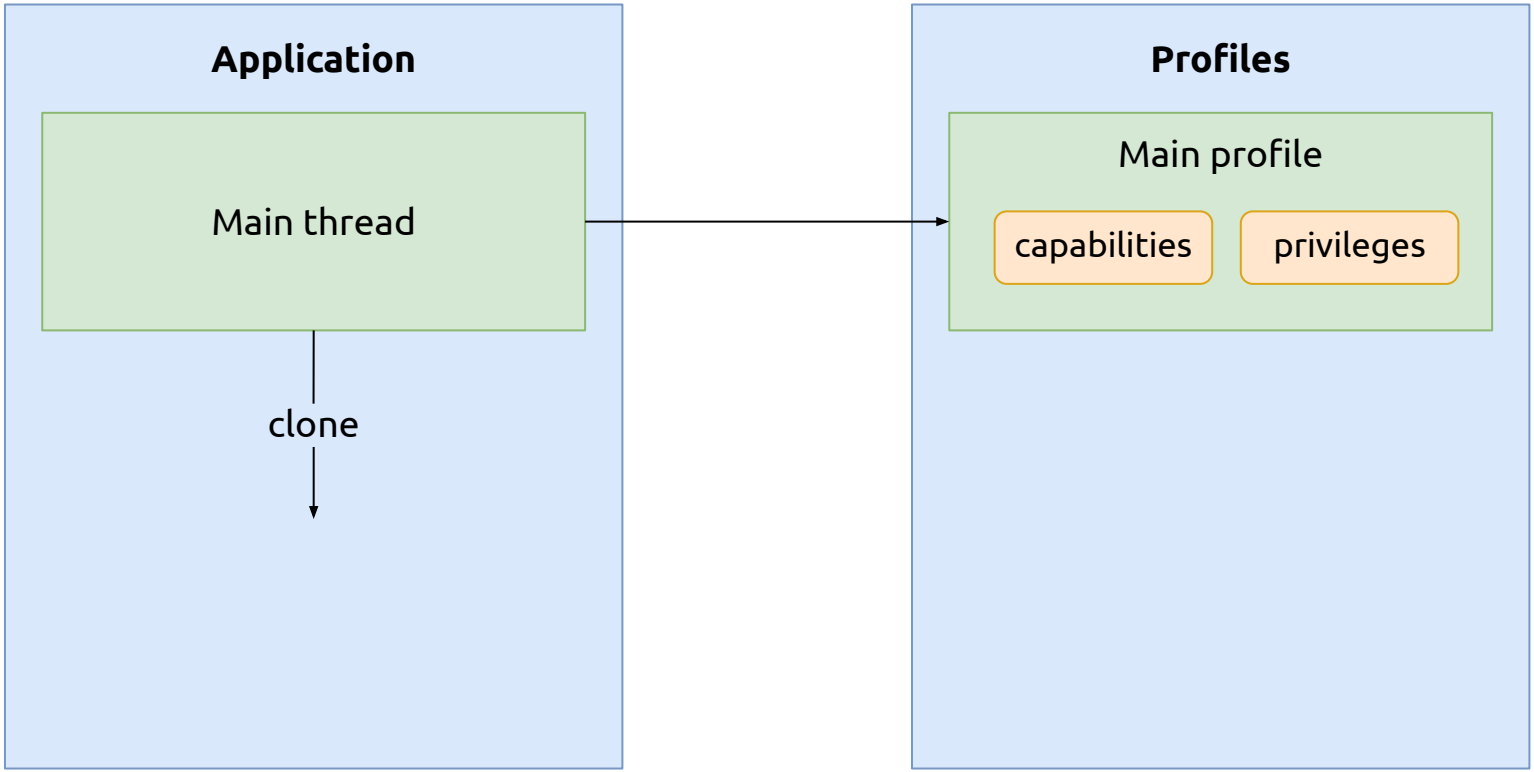
Conditional

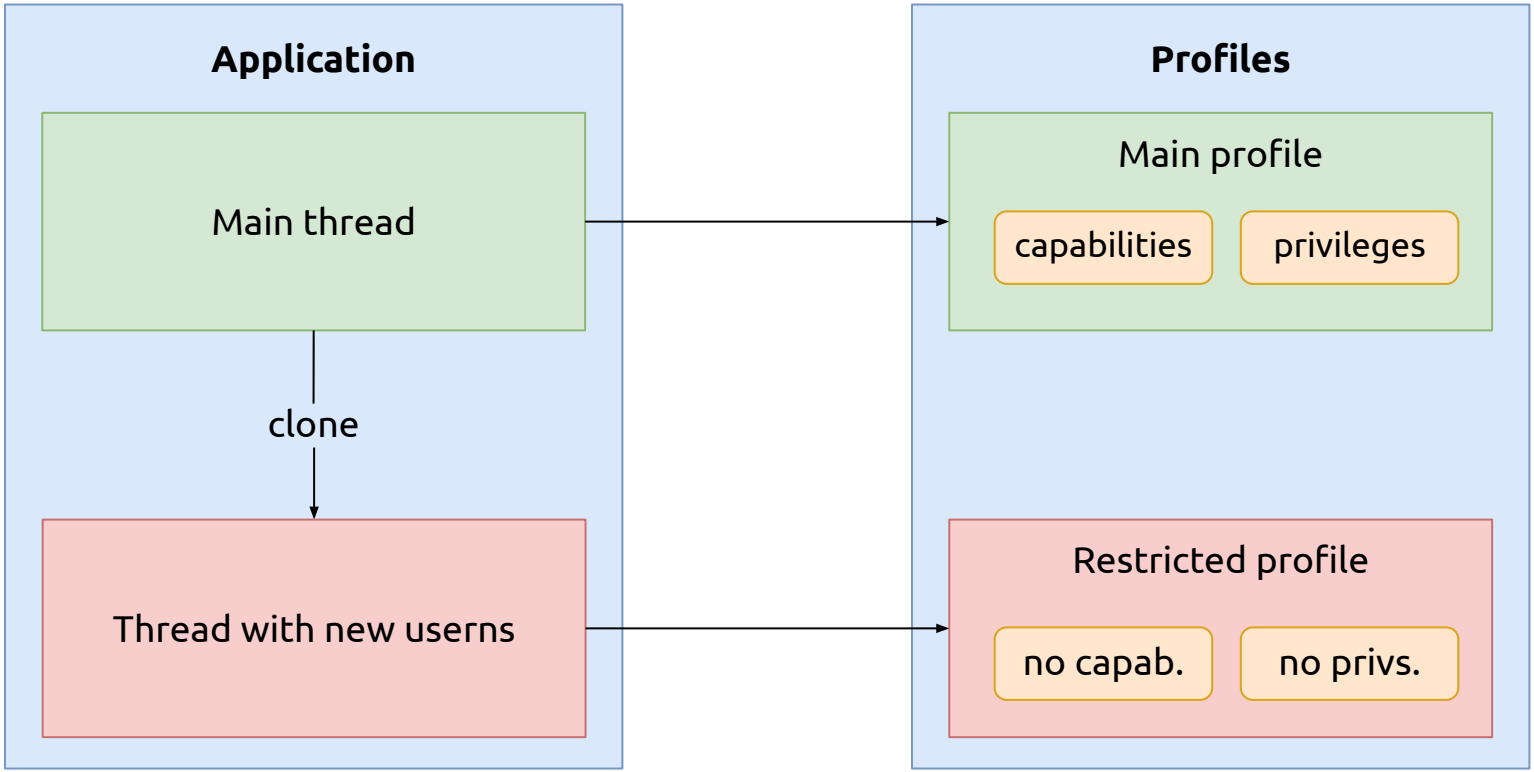
```
profile A {
     if (in userns) {
          capability net_admin,
          capability sys_admin,
          allow mount,
          …
     }

     allow userns create,
     allow unix peer=C,
     …
}
```

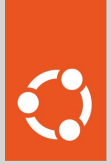
Transition

```
Profile A {
     allow userns create -> B,
     allow unix peer=C,
     …
}

profile B {
     allow capability net_admin,
     allow capability sys_admin,
     allow mount,
     …
}
```


Application
Main thread
Profiles
Main profile
capabilities
privileges


Application
Main thread
clone
Profiles
Main profile
capabilities
privileges


Application
Main thread
clone
Thread with new userns
Profiles
Main profile
capabilities
privileges
Restricted profile
no capab.
no privs.



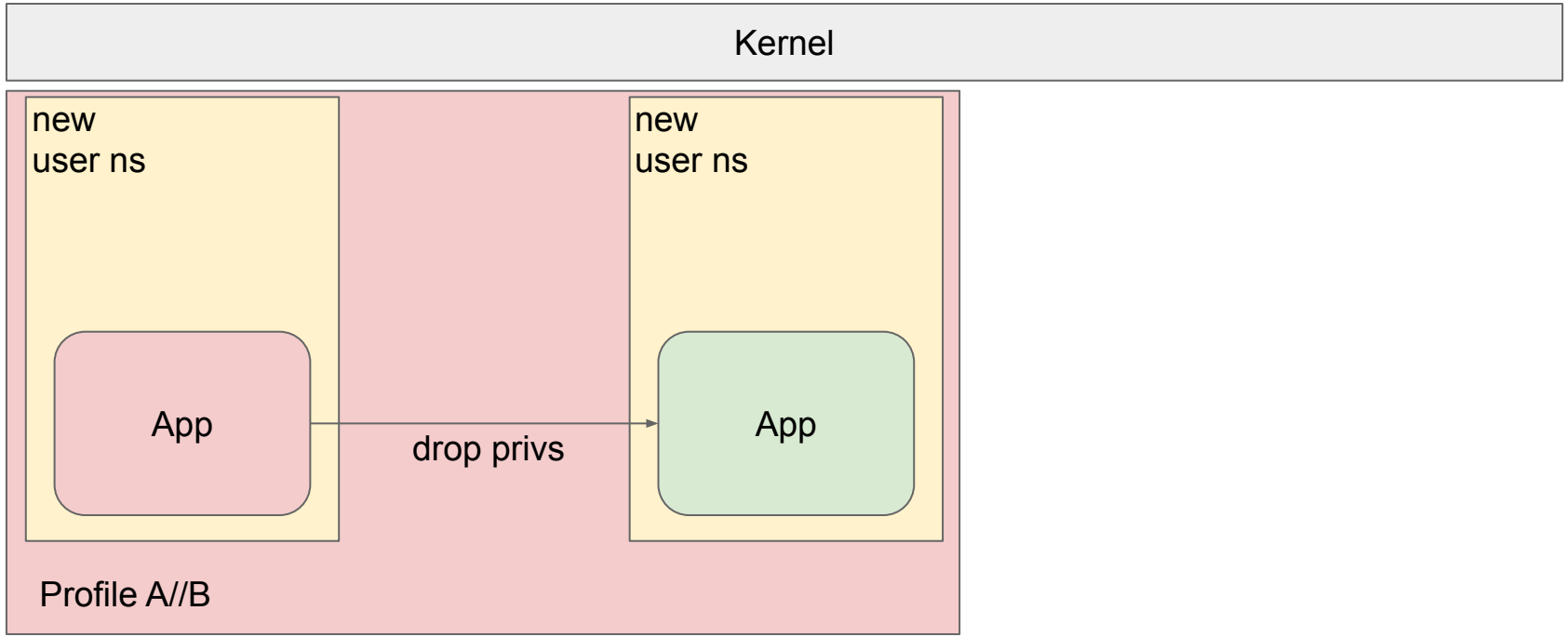
Sandboxing Apps

Sandbox apps

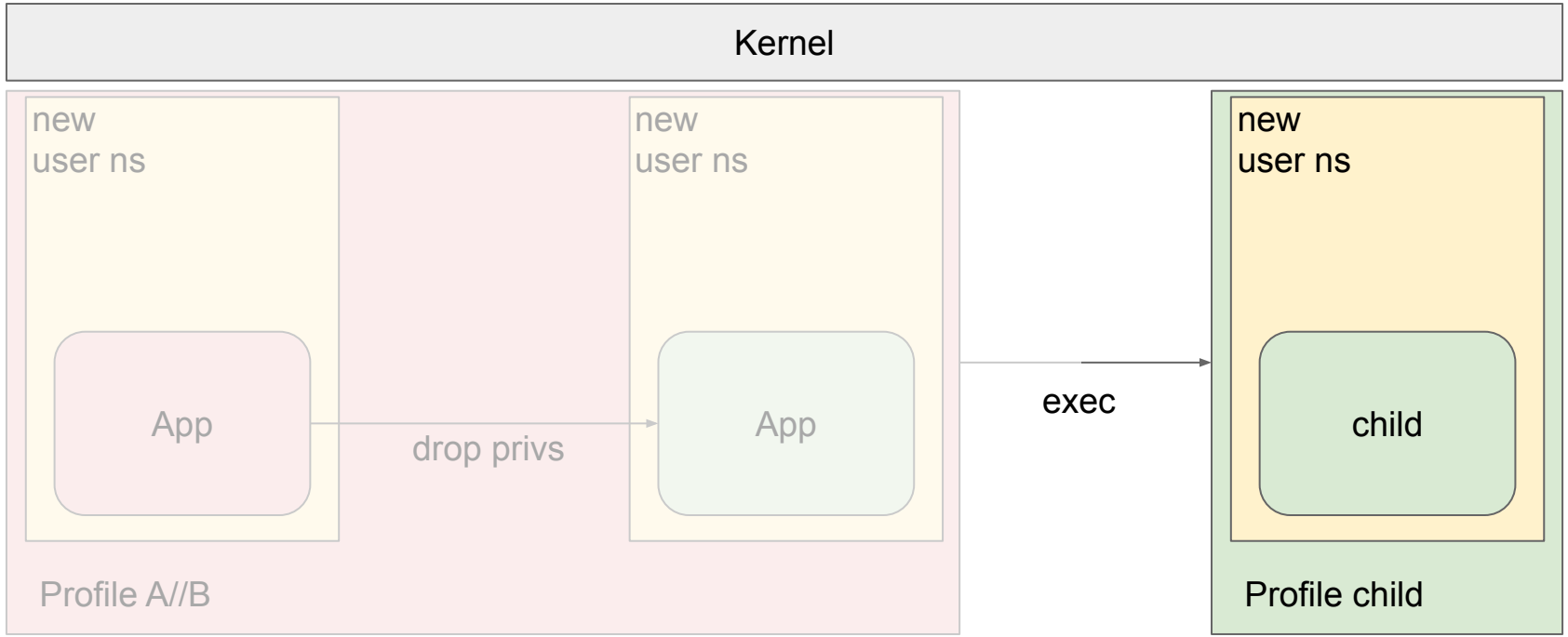
- Sandbox app:
  - Launches other applications into user namespaces

- Examples
  - unshare
  - bwrap
  - firejail
  - nsjail
  - snapd
  - flatpak

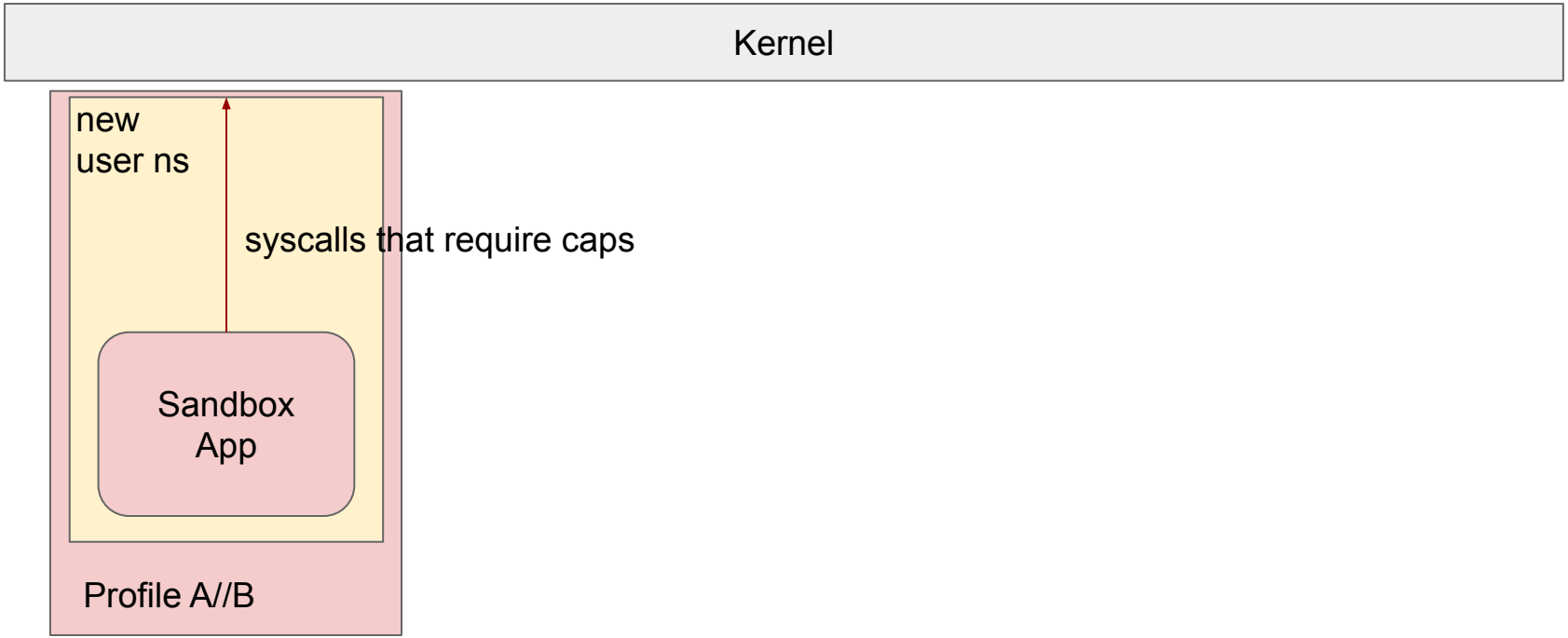
Sandbox launch App



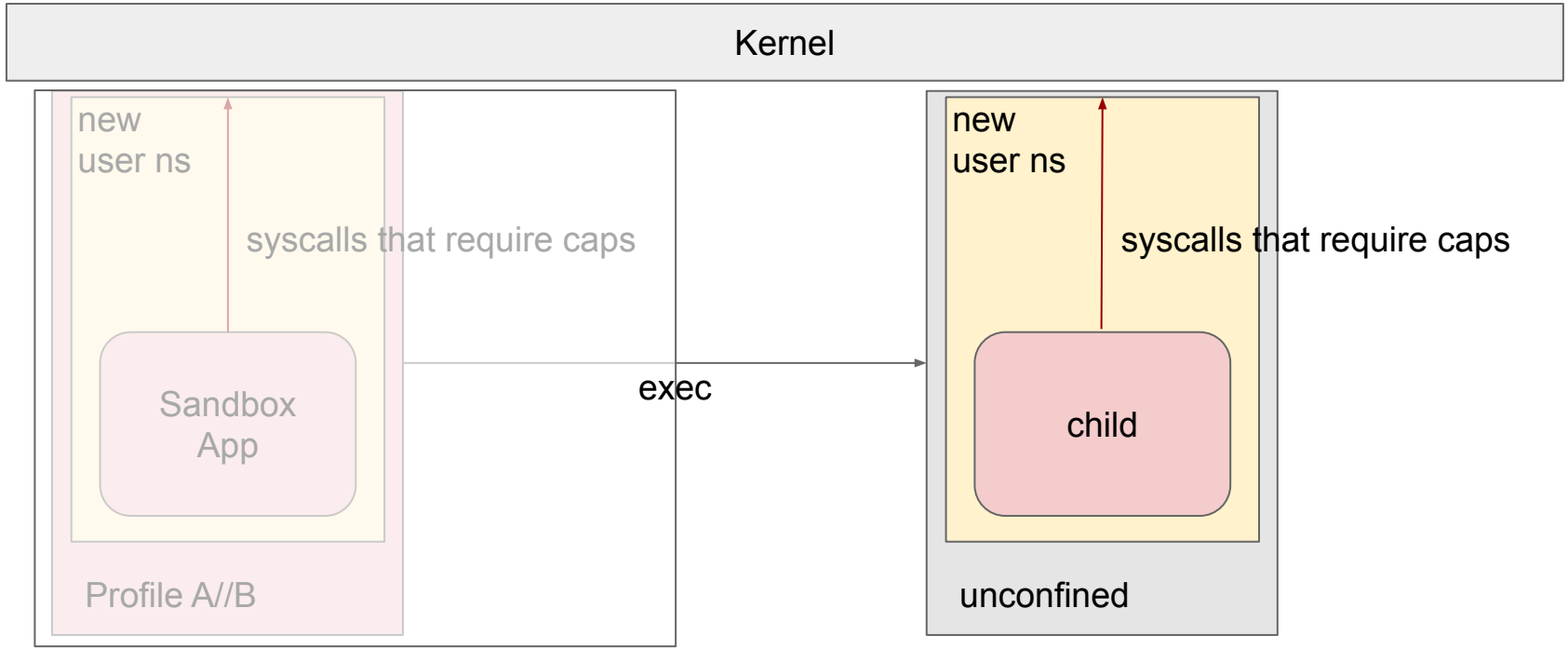
Sandbox with exec as a barrier



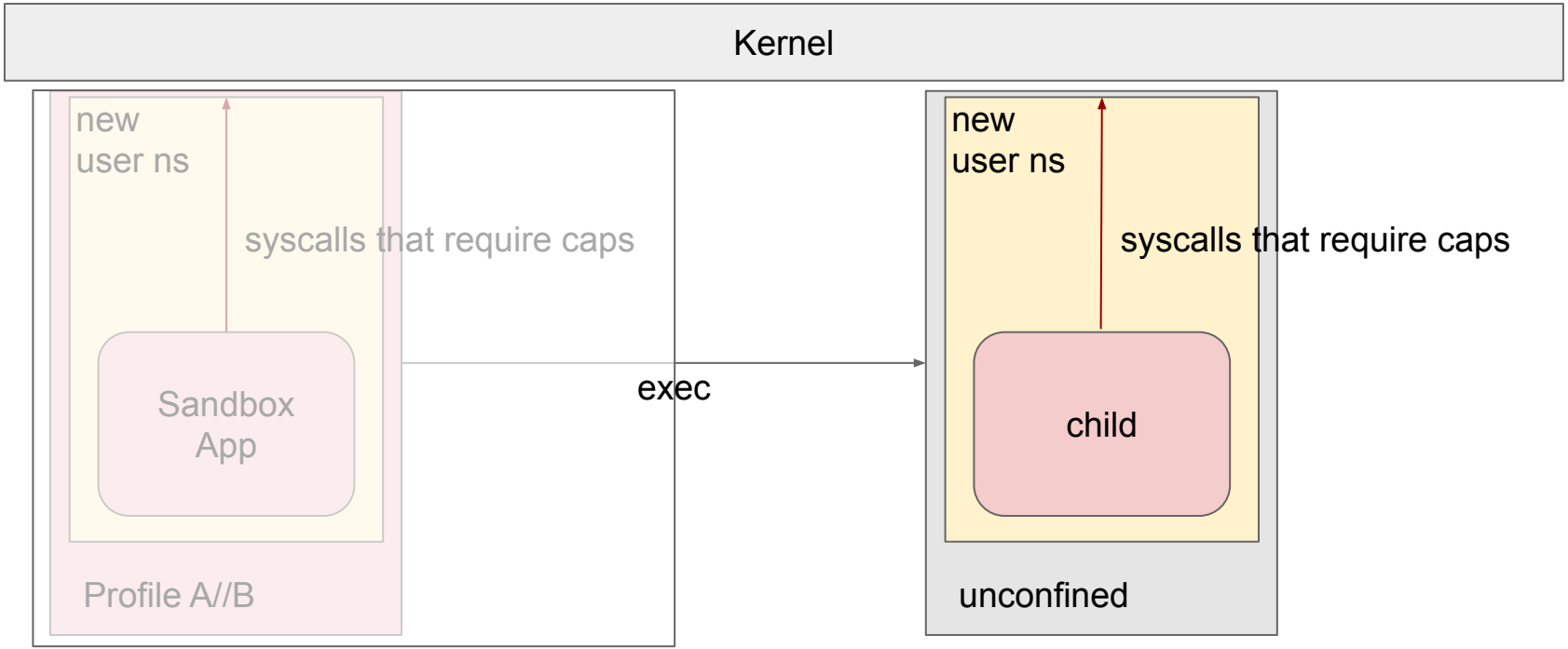
Sandbox exec - hand off privileges



Sandbox exec - hand off privileges



Unpriv User Namespace Restriction Escape!



Sandbox apps

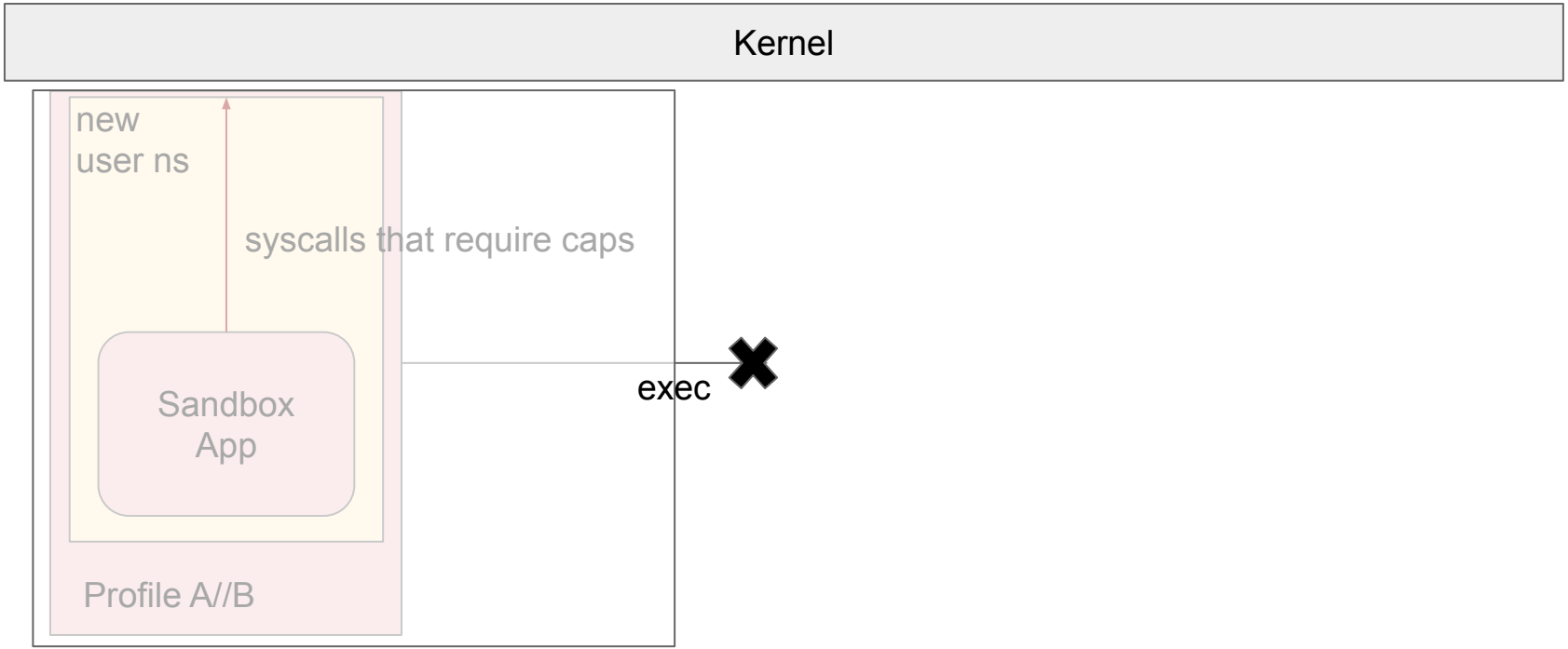
Must Be Confined

Sandbox apps

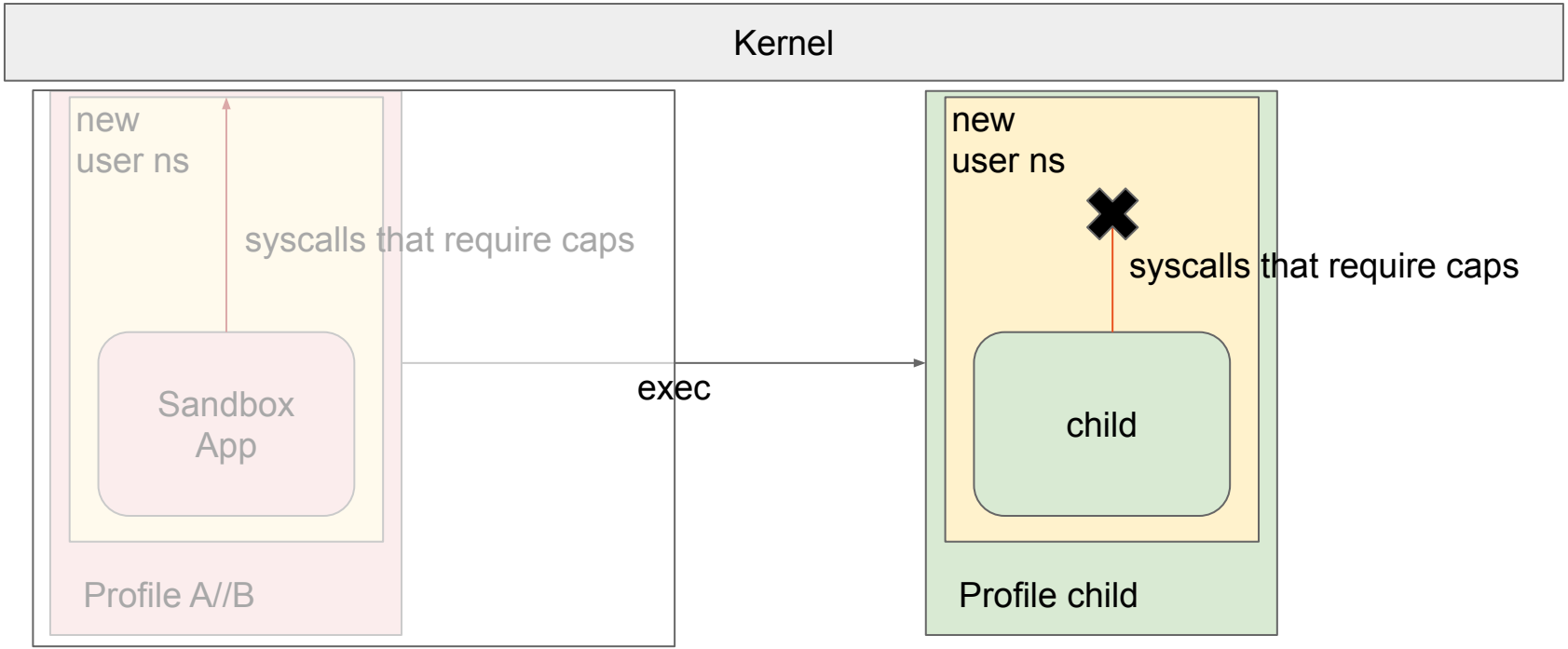
Must Be Confined

Children Must Be Confined

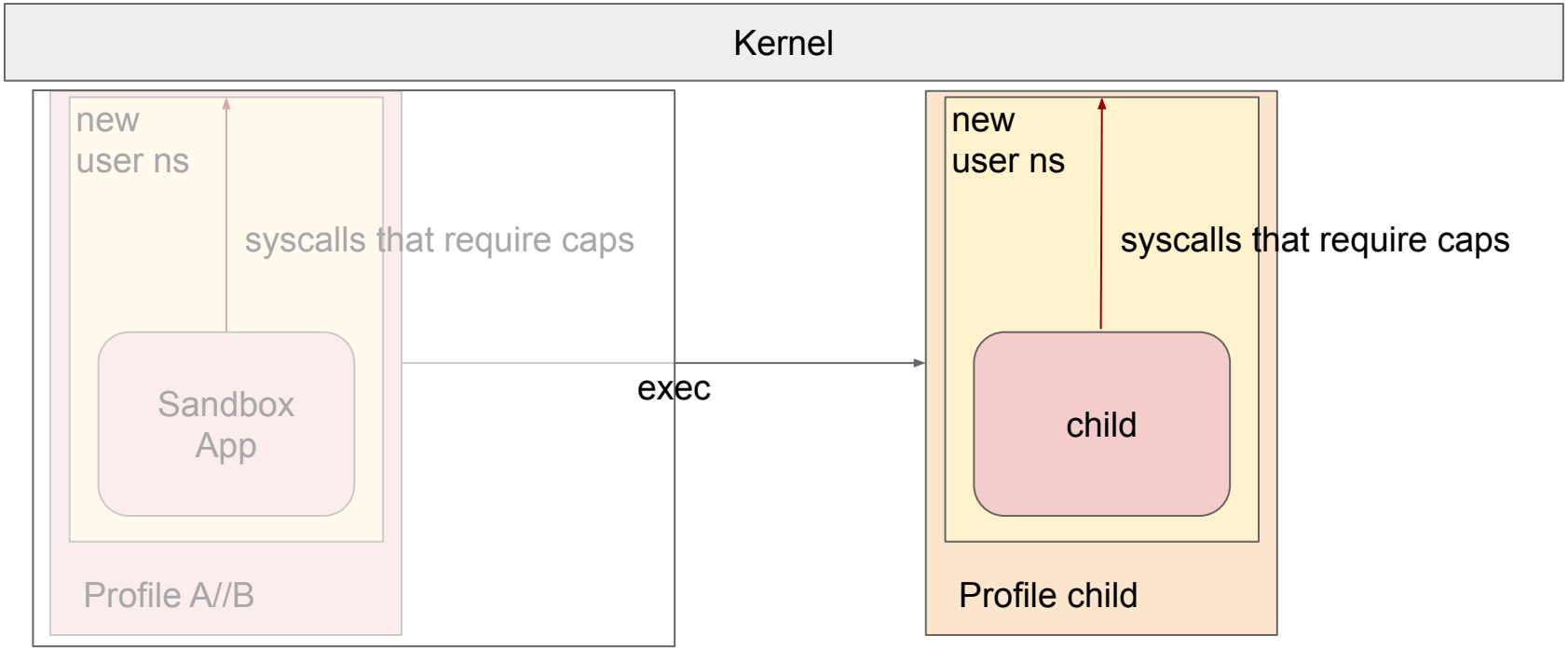
Disallow exec to unconfined



Profile Restricts Capabilities



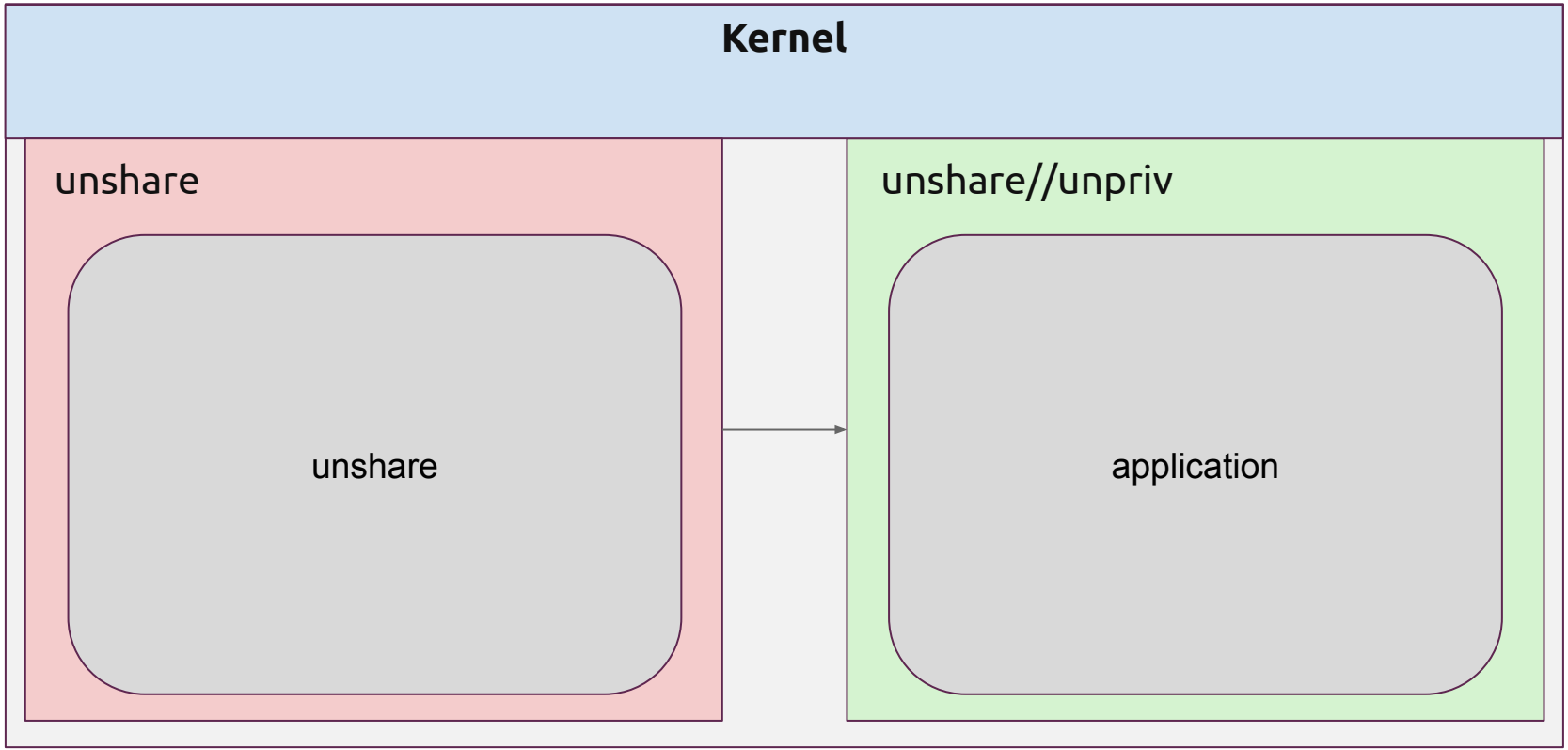
Special Case: Profile Allows Capabilities



2. External Sandboxing

```
abi <abi/4.0>,

include <tunables/global>

profile unshare /usr/bin/unshare flags=(attach_disconnected) {

  allow userns,
  allow capability,
  audit allow cx /** -> unpriv,
  allow network,
  allow unix,
  allow ptrace,
  allow signal,
  allow mqueue,
  allow io_uring,
  …
```

Unshare launching App



Bwrap and No_new_privileges

```
abi <abi/4.0>,

include <tunables/global>

profile bwrap /usr/bin/bwrap flags=(attach_disconnected) {
  allow capability,
  allow userns,
  allow px /** -> bwrap//&unpriv_bwrap,

  allow network,
  allow unix,
  allow ptrace,
  allow signal,
  allow mqueue,
  allow io_uring,
  …
```

Ubuntu
apparmor package

Apparmor: New update broke flatpak with `apparmor="DENIED"`

Noble (24.04) • **Bug #2072811**

Bug #2072811 reported by klo on 2024-07-11

This bug affects 20 people. Does this bug affect you? 110

Affects	Status	Importance	Assigned to	Milestone
apparmor (Ubuntu)	Status tracked in Oracular			
Noble	Fix Committed	Critical	Unassigned	Target to milestone
Oracular	Fix Released	High	Unassigned	Target to milestone

Also affects project | Also affects distribution/package | Target to series

Bug Description

The recent apparmor update appear to have broken some flatpak's ability to save file, e.g.:
- org.keepassxc.KeePassXC
- org.ksnip.ksnip

It seems update introduced a new profile ("/etc/apparmor.d/bwrap-userns-restrict"), which is causing the issue below.

**** To reproduce ****

(I'm using KeepassXC as example, but same issue for ksnip):

Report a bug

This report contains **Public** information

Everyone can see this information.

Mark as duplicate
Convert to a question
Link a related branch
Link to CVE
Change lock status

You have subscriptions that may cause you to receive notifications, but you are not directly subscribed to this bug's notifications.

Mute bug mail
Edit bug mail

Other bug subscribers

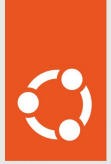
Subscribe someone else

Notified of all changes

Alex Garel

Breaking Flatpak

- treat bwrap differently when called from flatpak
  - Flatpak calls bwrap internally
  - But
    - Security policy dictated by flatpak
    - Not quite as bad as letting an attacker call bwrap directly
  - General principle
    - For several apps confinement might be different based on how it is called
- snapd more trusted than flatpak
  - Interface to use userns requires review



Issues

Issues

- Out of Archive
  - We Can ship policy for some well known Apps

Issues

- Out of Archive
  - We Can ship policy for some well known Apps
    - steam, firefox, …
    - but not all

Issues

- Out of Archive
  - We Can ship policy for some well known Apps
    - steam, firefox, …
    - but not all

- Applications from user writable locations

Issues

- Out of Archive
  - We Can ship policy for some well known Apps
    - steam, firefox, …
    - but not all

- Applications from user writable locations
  - Can not allow Applications from user writable locations to use userns

Issues

- Out of Archive
  - We Can ship policy for some well known Apps
    - steam, firefox, …
    - but not all

- Applications from user writable locations
  - Can not allow Applications from user writable locations to use userns
    - at least at the distro level
    - allow user to do it

Issues

- Out of Archive
  - We Can ship policy for some well known Apps
    - steam, firefox, …
    - but not all

- Applications from user writable locations
  - Can not allow Applications from user writable locations to use userns
    - at least at the distro level
    - allow user to do it
  - AppImages

Issues

- Out of Archive
  - We Can ship policy for some well known Apps
    - steam, firefox, …
    - but not all

- Applications from user writable locations
  - Can not allow Applications from user writable locations to use userns
    - at least at the distro level
    - allow user to do it
  - AppImages
    - Electron based apps

Issues

- Out of Archive
  - We Can ship policy for some well known Apps
    - steam, firefox, …
    - but not all

- Applications from user writable locations
  - Can not allow Applications from user writable locations to use userns
    - at least at the distro level
    - allow user to do it
  - AppImages
    - Electron based apps

- Upstreams circumventing the restriction

Policy to bypass the restriction

Add apparmor rule to fix thumbnails

Code

Open **Sergio Costas** requested to merge `Sergio.Costas/gnome-shel…` into `ubuntu/master` 17 hours ago

Overview 2 | Commits 1 | Pipelines 0 | Changes 2

Desktop Icons NG uses libgnome-shell to generate thumbnails. Internally, it uses bwrap to execute the thumbnailer inside a container. Unfortunately, this no loger works directly due to the security changes done in Ubuntu 23.10, which disables, by default, user namespaces, and requires specific rule files for each application that needs it.

More info in https://ubuntu.com/blog/ubuntu-23-10-restricted-unprivileged-user-namespaces

This patch adds an apparmor exception file to ensure that it works. Without it, the thumbnailer just fails.

👍 0 👎 0

Members who can merge are allowed to add commits.

Approval is optional

Ready to merge by members who can write to the target branch.

- **1** commit will be added to `ubuntu/master`.
- Source branch will be deleted.

0 Assignees
None

0 Reviewers
None

Labels
None

Milestone
None

Time tracking
No estimate or time spent

2 Participants

Activity

All activity

Issues

- Out of Archive
  - We Can ship policy for some well known Apps
    - steam, firefox, …
    - but not all

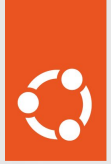
- Applications from user writable locations
  - Can not allow Applications from user writable locations to use userns
    - at least at the distro level
    - allow user to do it
  - AppImages
    - Electron based apps

- Upstreams circumventing the restriction

- Users don’t understand

Easy - Escape Hatch

```
# This profile allows everything and only exists to give the
# application a name instead of having the label "unconfined"

abi <abi/4.0>,
include <tunables/global>

profile runc /usr/sbin/runc flags=(unconfined) {
  userns,

  # Site-specific additions and overrides. See local/README for details.
  include if exists <local/runc>
}
```



Dealing with AppImages & out of tree

Enable through unconfined profile

```
# This profile allows everything and only exists to give the
# application a name instead of having the label "unconfined"

abi <abi/4.0>,
include <tunables/global>

profile runc /usr/sbin/runc flags=(unconfined) {
  userns,

  # Site-specific additions and overrides. See local/README for details.
  include if exists <local/runc>
}
```

Enable through unconfined profile

- For average user
  - Unexpected crash
  - Unclear that the crash is caused by User Namespace
  - Unfamiliar with AppArmor

- Very confusing for the end user

- Not a practical solution for most users

User confusion

```
user@user:~/my-app$ npm start

> my-app@1.0.0 start
> electron-forge start
```

Enable through unconfined profile

- Goal: If a user namespace is denied
  - User-friendly notification
  - The user understands what happened
  - Can allow/deny userns creation for this program.

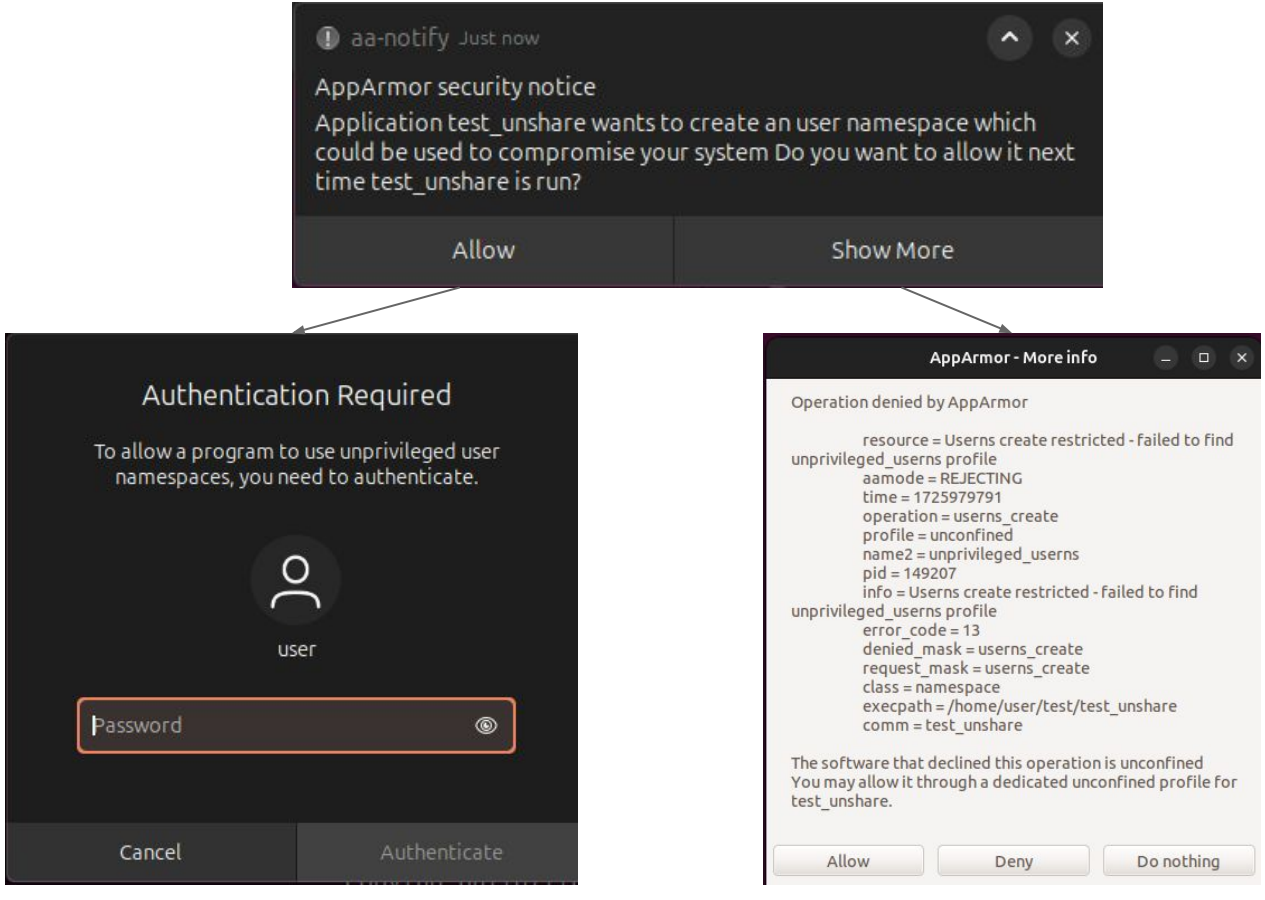
- Constraints:
  - Easy to understand
  - No flooding of notifications
  - Not abusable by malicious users.
  - Well integrated GUIs

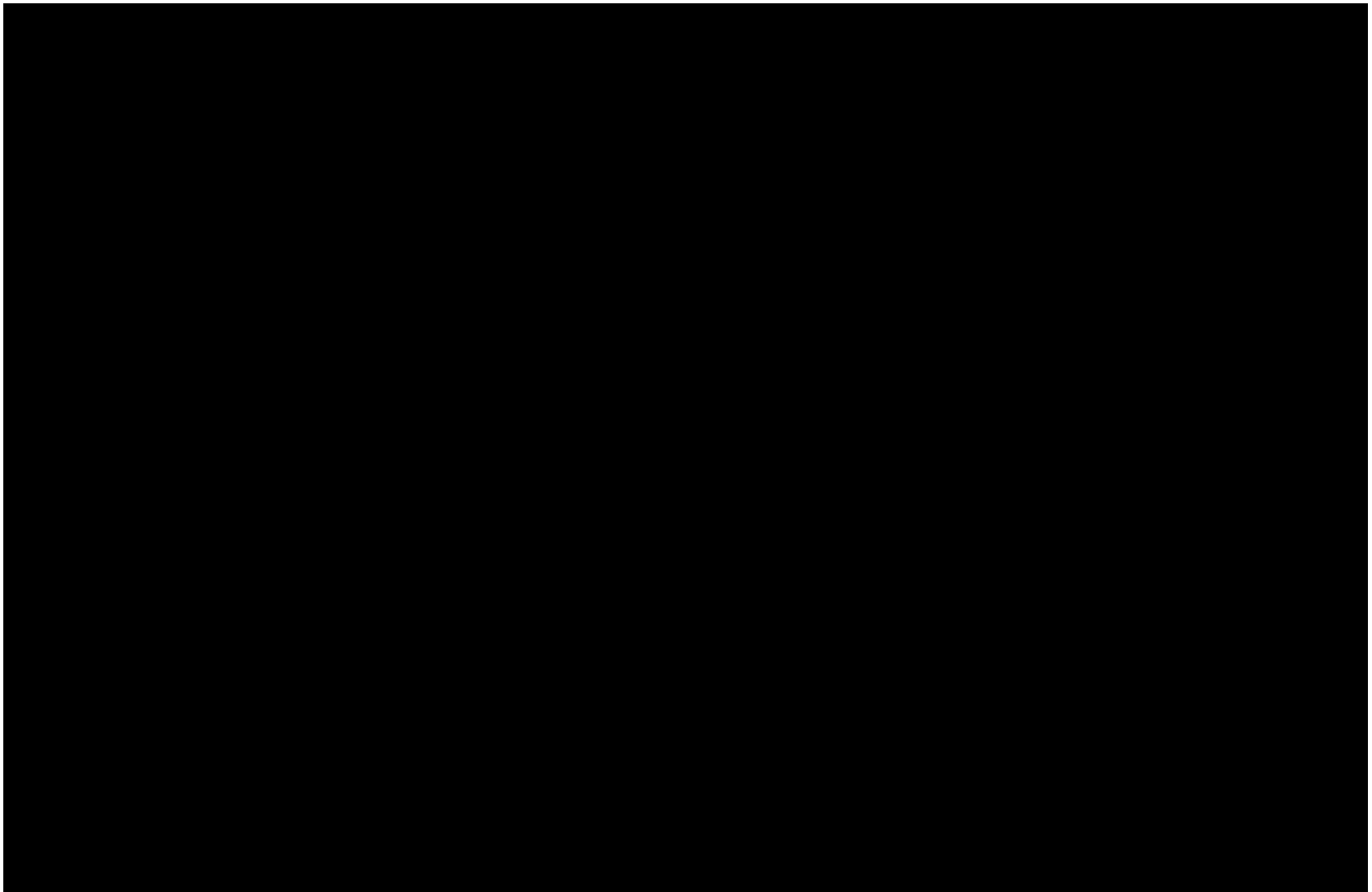
Enablement through aa-notify

- aa-notify:
  - Background task
  - Show denied userns creation
  - User-friendly GUIs
  - The user can make actions.
  - Themable interfaces
  - Easily integrable in distributions

- Useful options:
  - `--poll`
  - `--filter.operation=userns_create`
  - `--prompt-filter userns`

Implementation: aa-notify





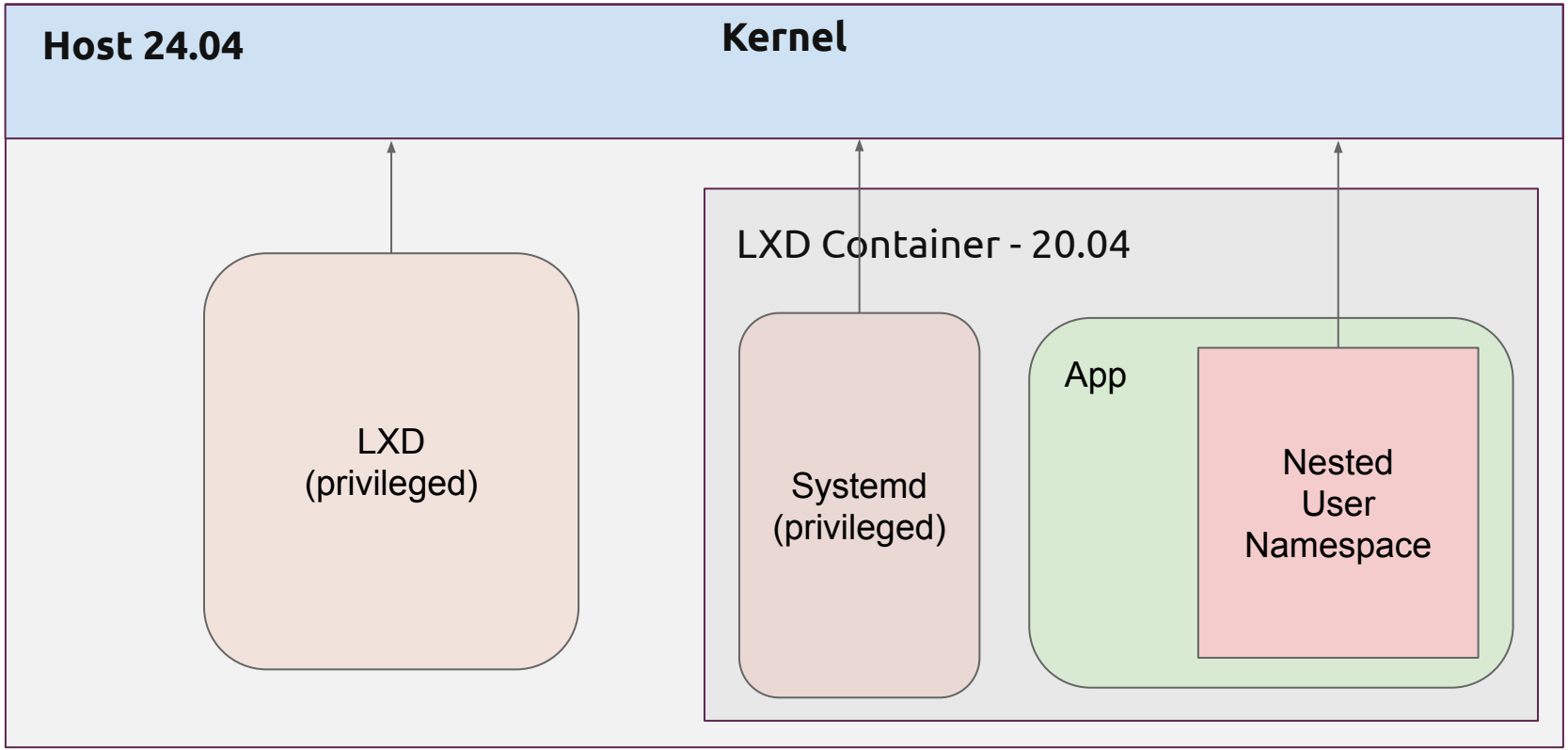
LXD

Like a Sandboxing App on steriods

LXD

- Disables userns restriction globally
  - Can’t be done at system profile level
  - Container has a policy namespace
    - Can have its own policy
    - Don’t want restriction applied in old containers
- LXD containers can be used to by-pass the restriction
  - Not installed by default
  - User not in LXD group by default

24.04 Host


Host 24.04
Kernel
LXD
(privileged)
LXD Container - 20.04
Systemd
(privileged)
App
Nested
User
Namespace

More on the restriction

- Available in the kernel but disabled at boot

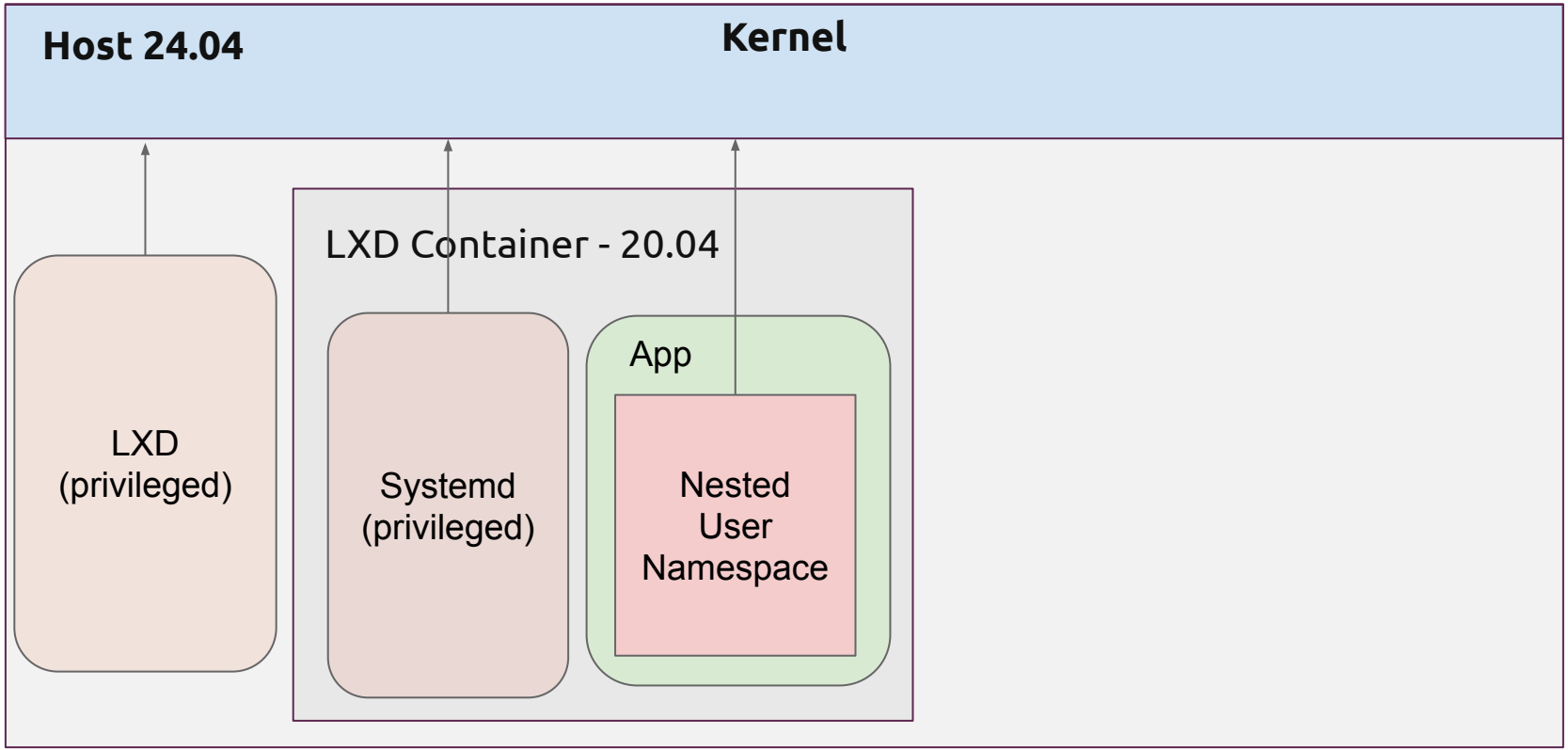
More on the restriction

- Available in the kernel but disabled at boot
- Enabled at boot by setting global policy boolean
  - Part of 24.04 boot, not 23.10 …

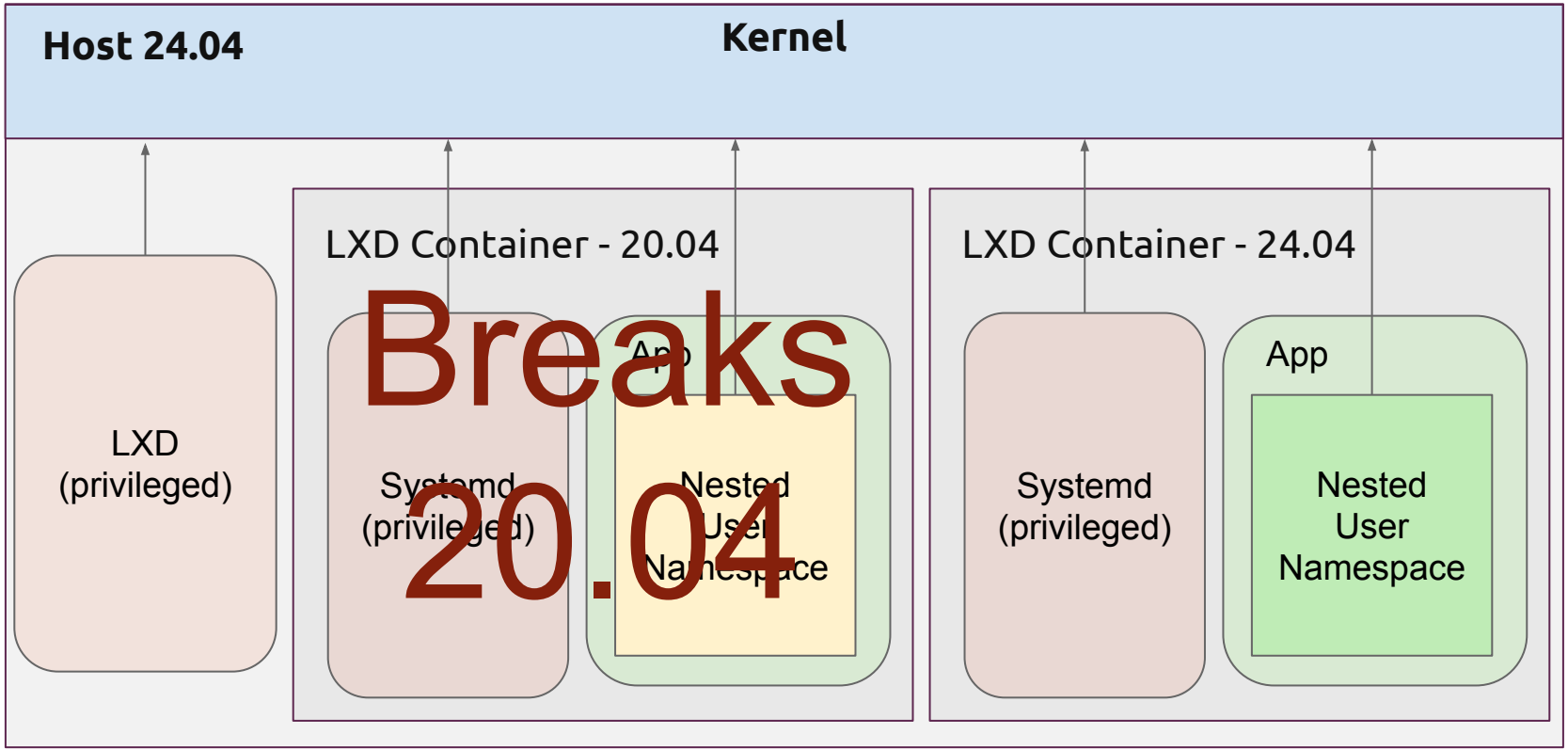
More on the restriction

- Available in the kernel but disabled at boot
- Enabled at boot by setting policy boolean
  - Part of 24.04 boot, not 23.10 …
- Can be toggled off/on at run time
  - LXD toggles it off

24.04 Host


Host 24.04
Kernel
LXD Container - 20.04
LXD
(privileged)
Systemd
(privileged)
App
Nested
User
Namespace

24.04 Host - Broken Guests

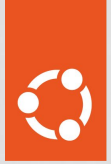


LXD

HUH

More on the restriction

- Available in the kernel but disabled at boot
- Enabled at boot by setting policy boolean
  - Part of 24.04 boot, not 23.10 …
- Can be toggled off/on at run time
  - LXD toggles it off
  - 24.04 Guest will turn it back on
    - Breaking 22.04 Guest

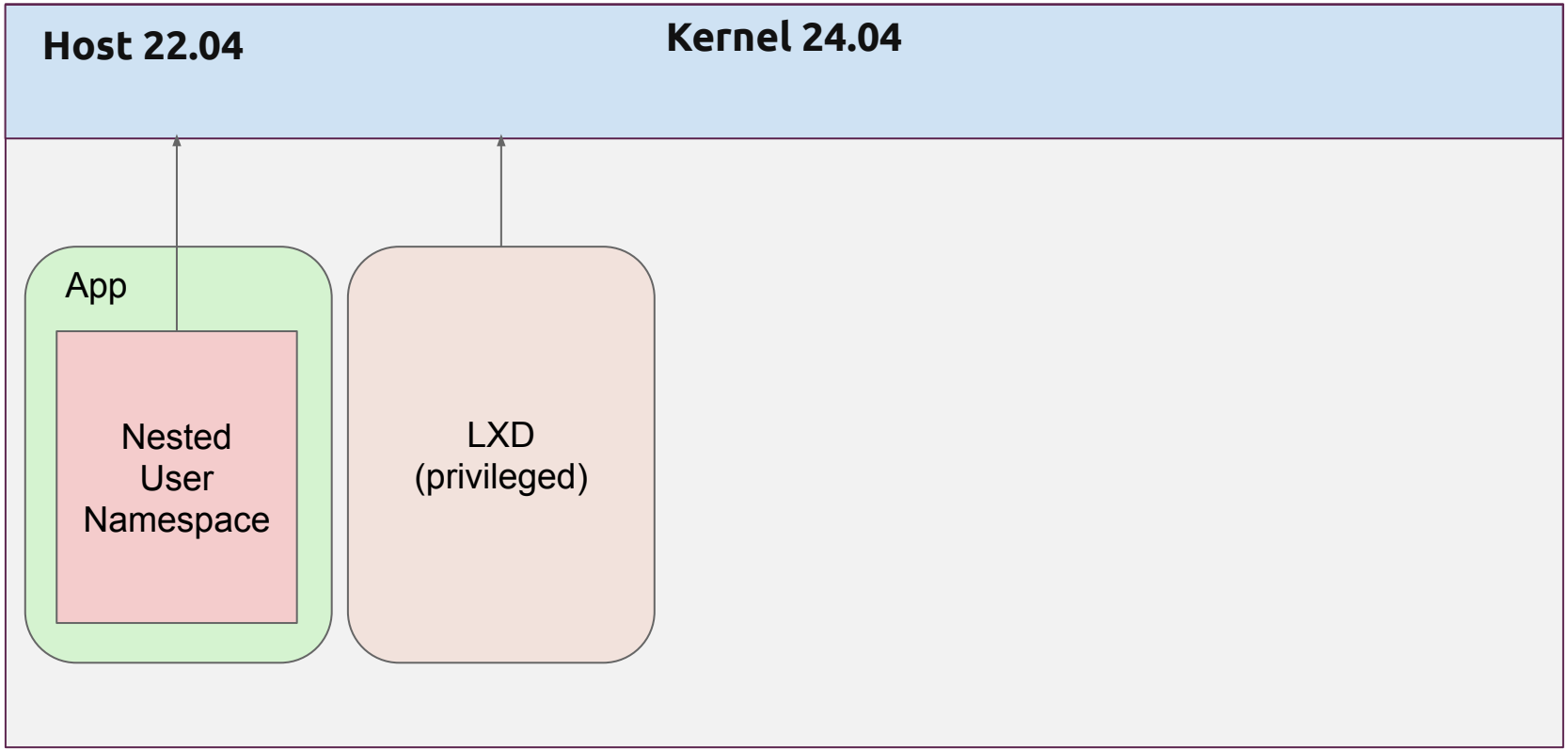


More LXD “Fun”

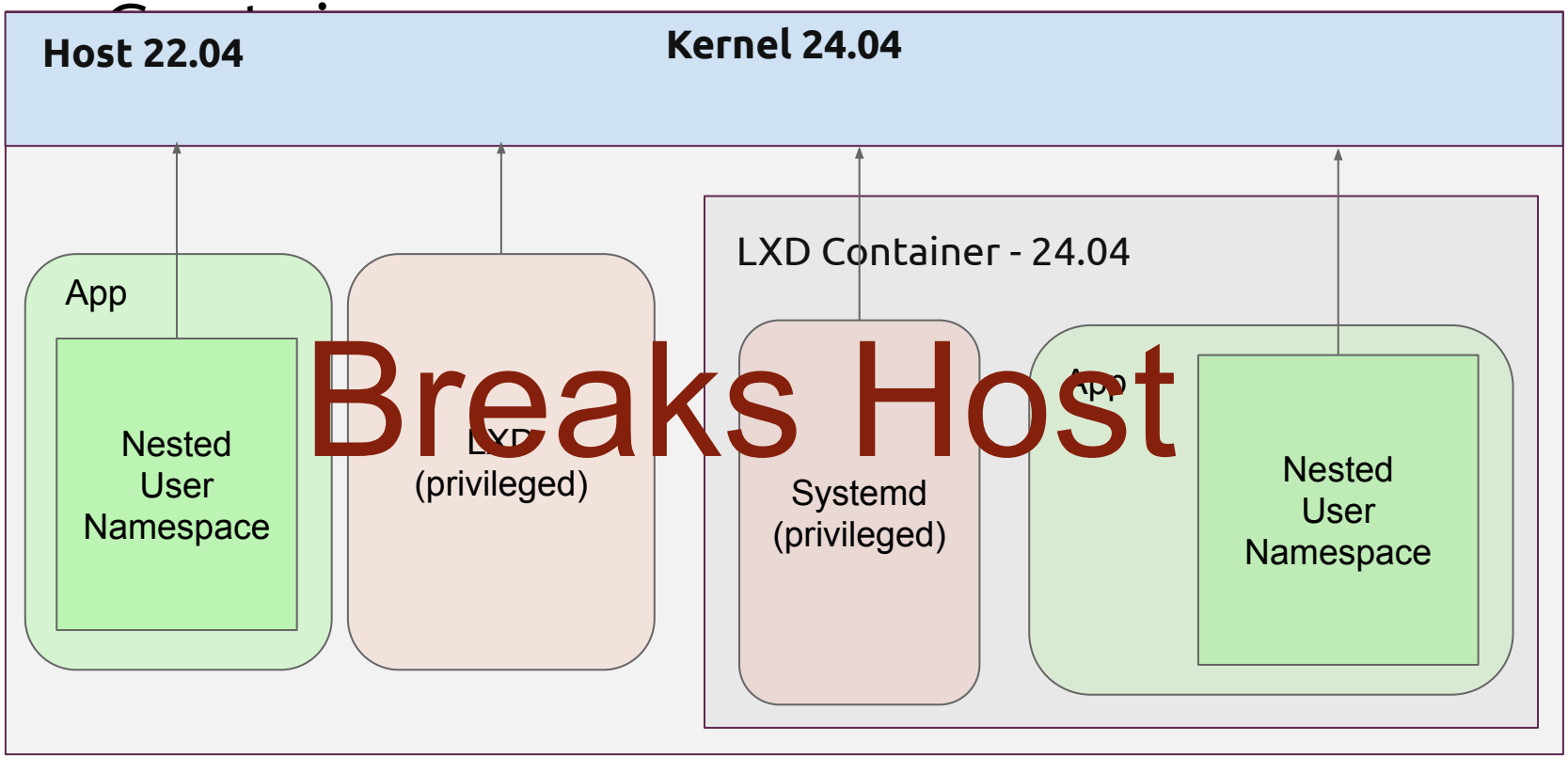
New Kernels on Old Releases

- 24.04 Kernel on 22.04
  - Restriction available in kernel
  - 22.04 does NOT enable

22.04 w/ 24.04 Kernel



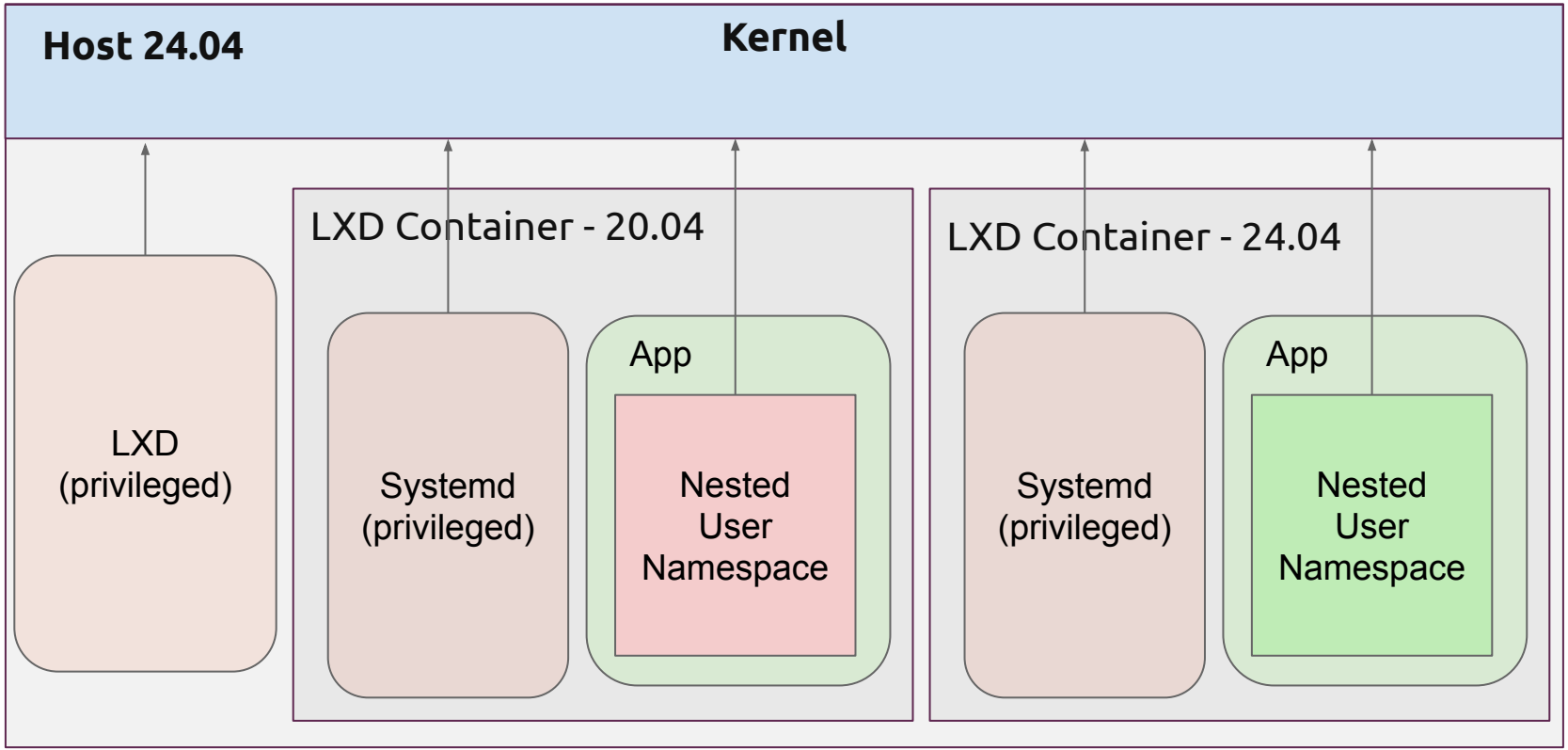
22.04 w/ 24.04 Kernel x 24.04



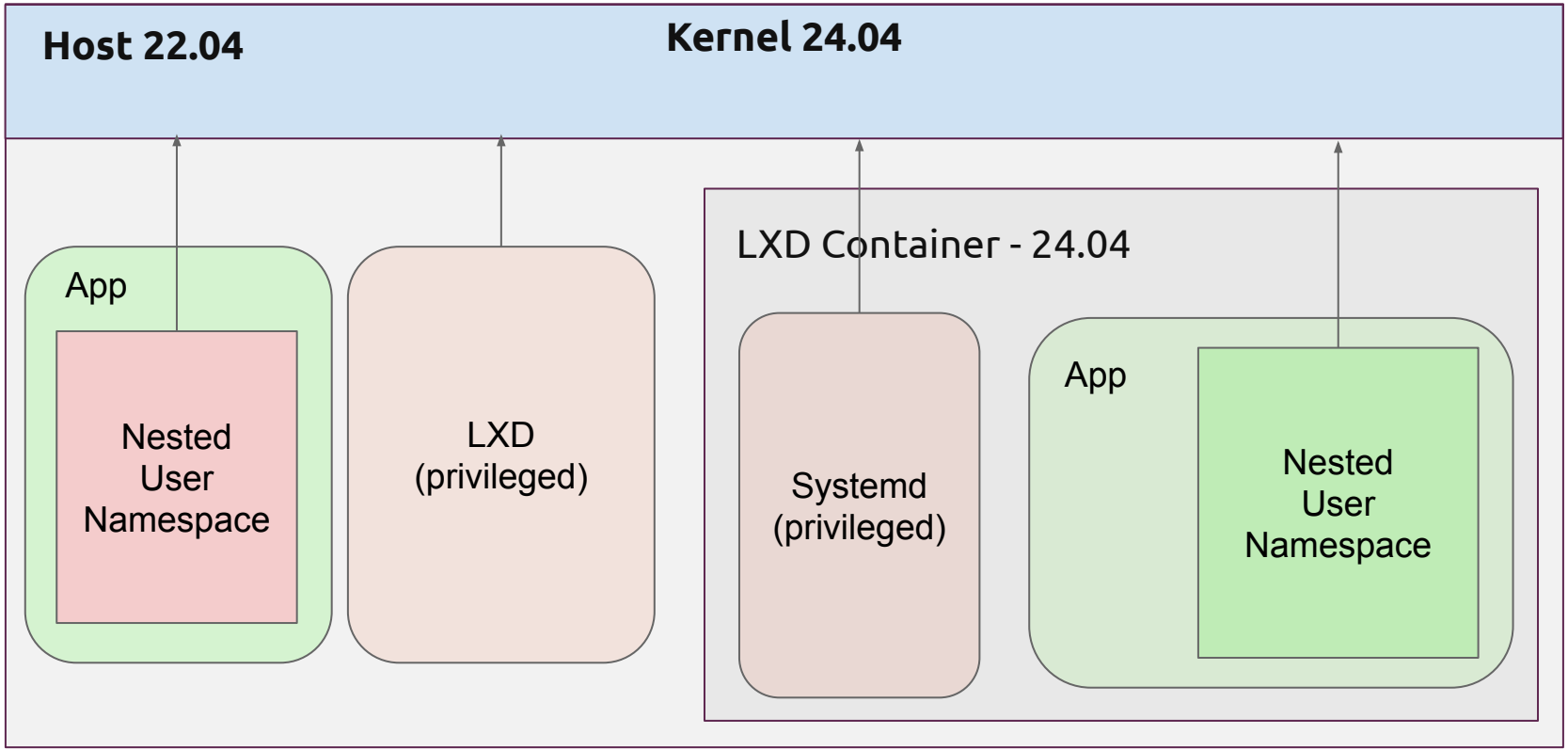
Virtualizing the Restriction

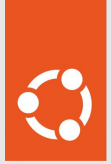
- Virtualize userns restriction
  - AppArmor Policy Namespace Level
  - Requires Container Manager
    - Setup policy namespace per container
  - Container
    - Enables restriction for its policy

24.04 Host - Virtualized



22.04 w/ 24.04 Kernel x 24.04 Container





Summary

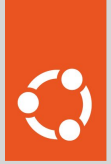
Userns “Trusted” Application

- Over 120 applications given access to userns
  - Most packaged with base policy
    - Some for Out of Archive
      - steam, firefox, chrome
      - …
  - Several carry profile in package
    - KDE
      - plasma, konqueror
      - …
- Flatpaks currently treated as 1 in count
- Several snap applications

Summary

- User namespaces
  - Cause security issues
  - Restricted in Ubuntu
  - … But needed by many applications

- In Ubuntu
  - Userns namespaces restricted
  - Trusted application can enable userns and capabilities
  - unprivileged_userns profile and tools to make this more user-friendly



Questions?