Pass
the SALT

Zero Dependencies sounds great...
until you try to share your code for the security good.

Eddie Billoir - PhD - Cybersecurity Architect
LeChatP
01/07/26

IRIT

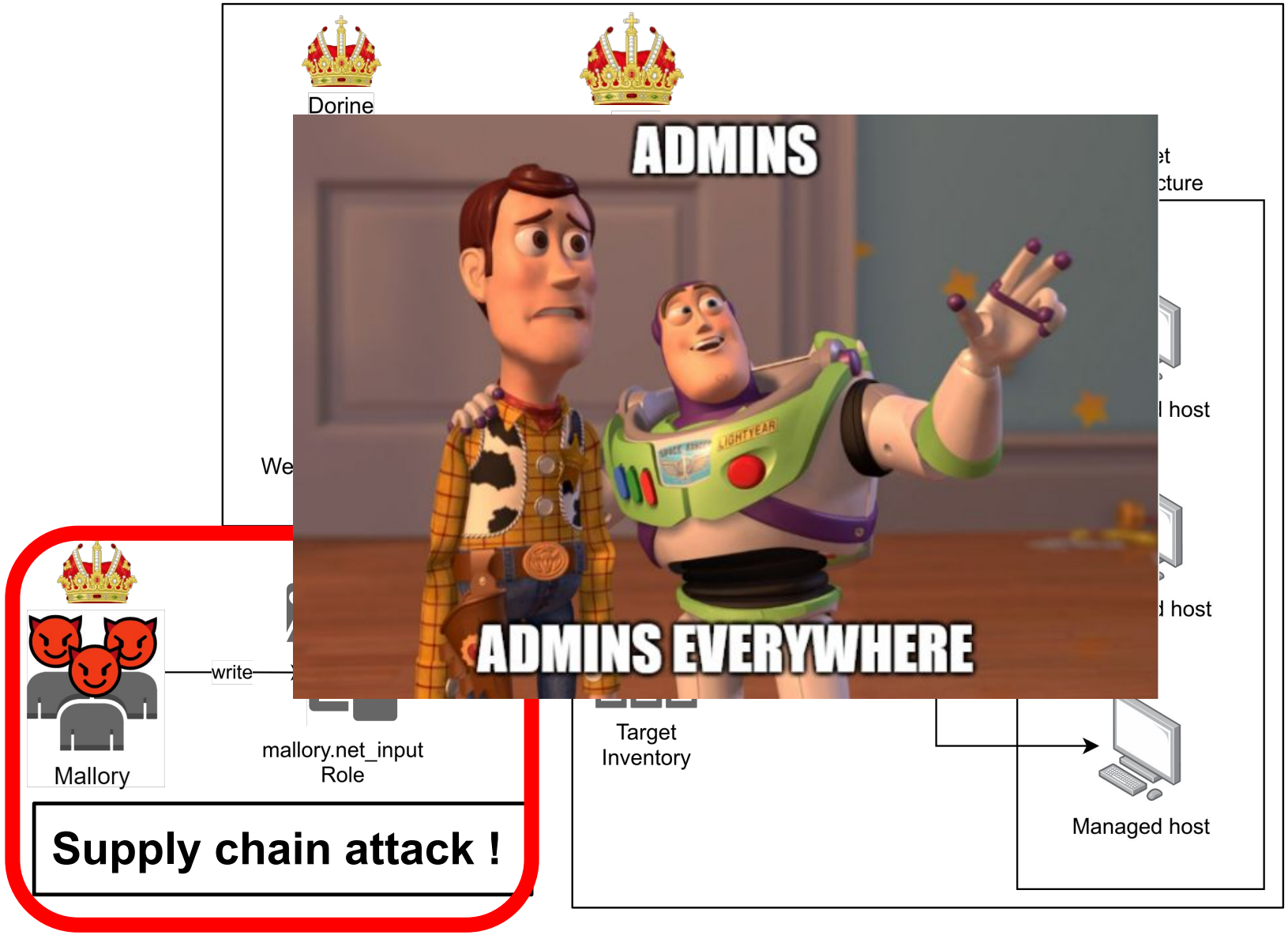
AIRBUS

Last Year…

AIRBUS

The administrator risk

Last Year…


Dorine
ADMINS
ADMINS EVERYWHERE
write
Mallory
mallory.net_input
Role
Target
Inventory
Managed host
Supply chain attack !

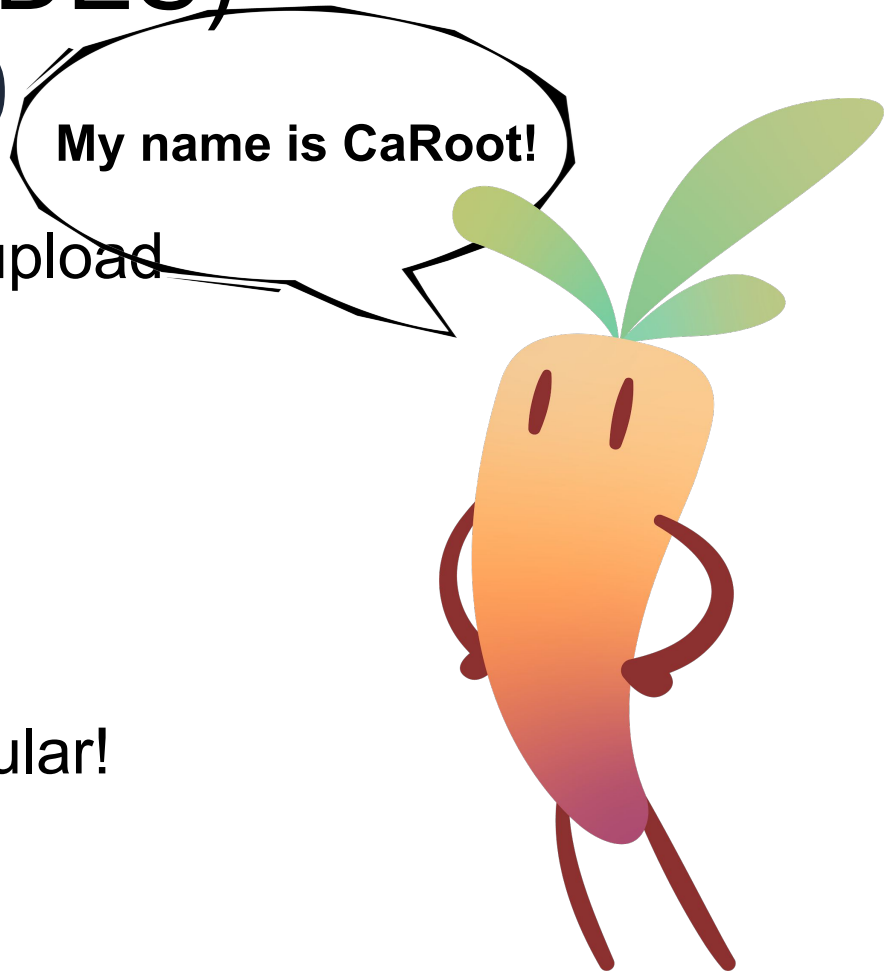
RootAsRole project, an alternative to sudo(-rs)

Root
As Role

Eva La Fougere

(RUMP SESSION SLIDES)

What is new and coming in 4.0



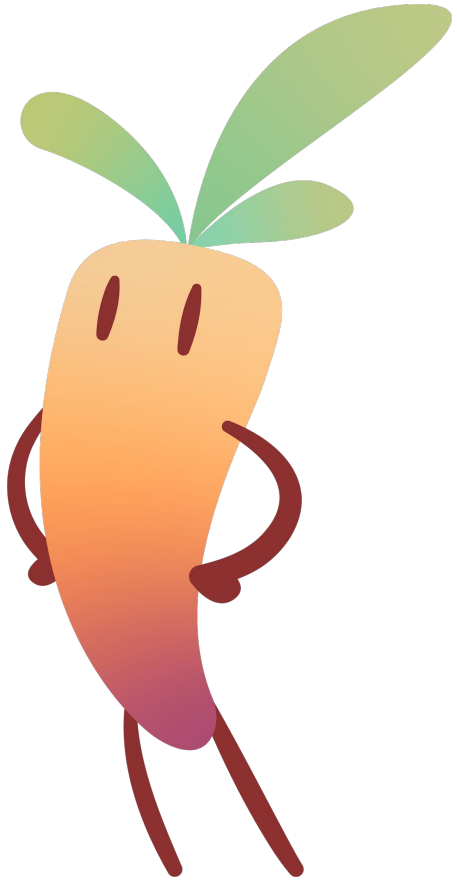
- Packaged for Debian in 2025, waiting for sponsor upload
- GdR Accessit Artefact PhD Reward :
  - https://github.com/LeChatP/RootAsAnsible
- Updating to Rust 2024 Edition! Finally!
- Enforcing Pedantic Clippy Rust
- New Crate! “rootasrole-exec”
  - Implements the sudo-rs architecture… but modular!
  - Manages Monitoring process ‘Proxy’ feature
  - Pty, Terminal, signals managed!
  - Orchestrating preexec steps : sid → namespace → pty→ filesystem → limits → umask → priv_drop → caps → fd_cleanup → signals → lockdown.

Choose what you want!

(RUMP SESSION SLIDES)

What is new and coming in 4.0

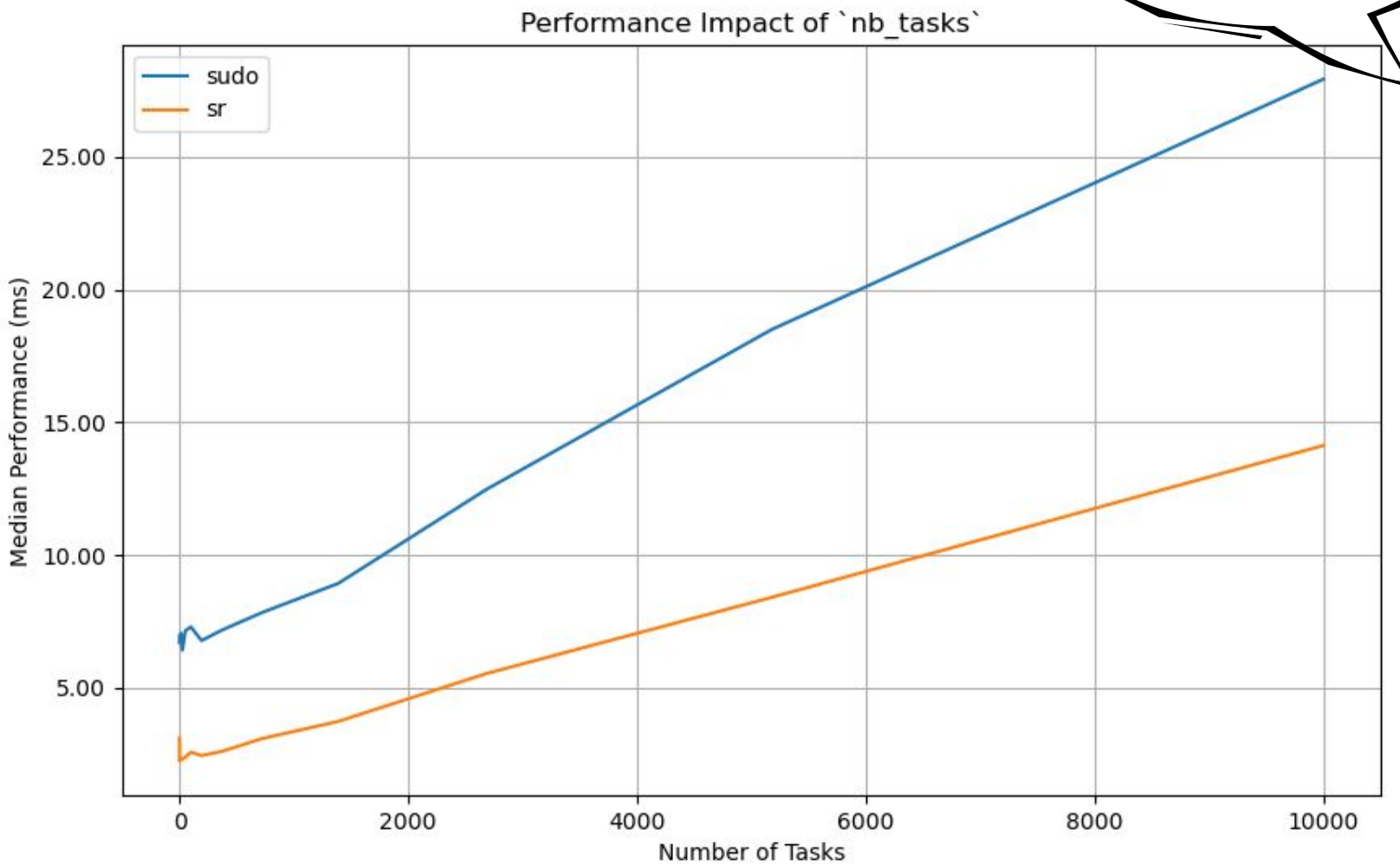
- “Secured” policy editor with vim or nano
- Folder based policy reading (/etc/security/rootasrole.d/)
- Umask is now in the policy…
- Stabilizing the “Rusty” compilation/installation process
- Landlock policy management!
- Added an AI letter on usage history, and policy for more transparency

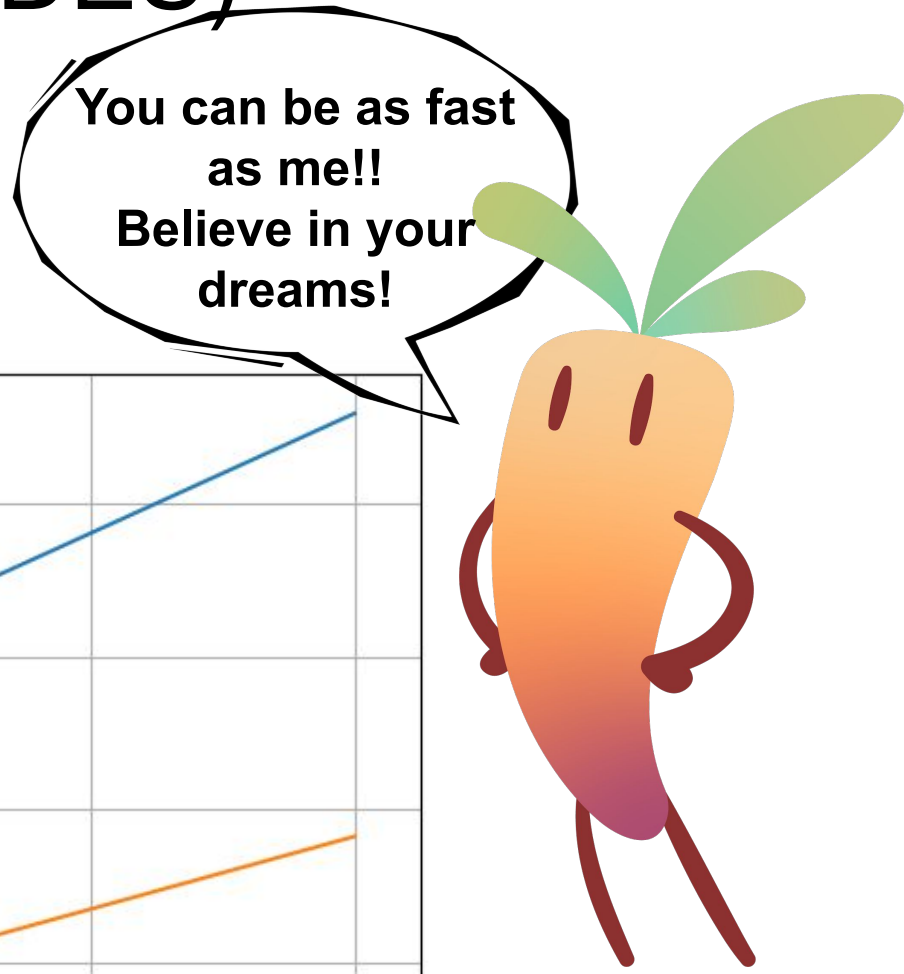


(RUMP SESSION SLIDES)

What is new and coming 4.0

⚡ **77% faster** than sudo when using a single rule
📈 **Scales 40% better** than sudo as more rules are added





AIRBUS

And now, what's next?

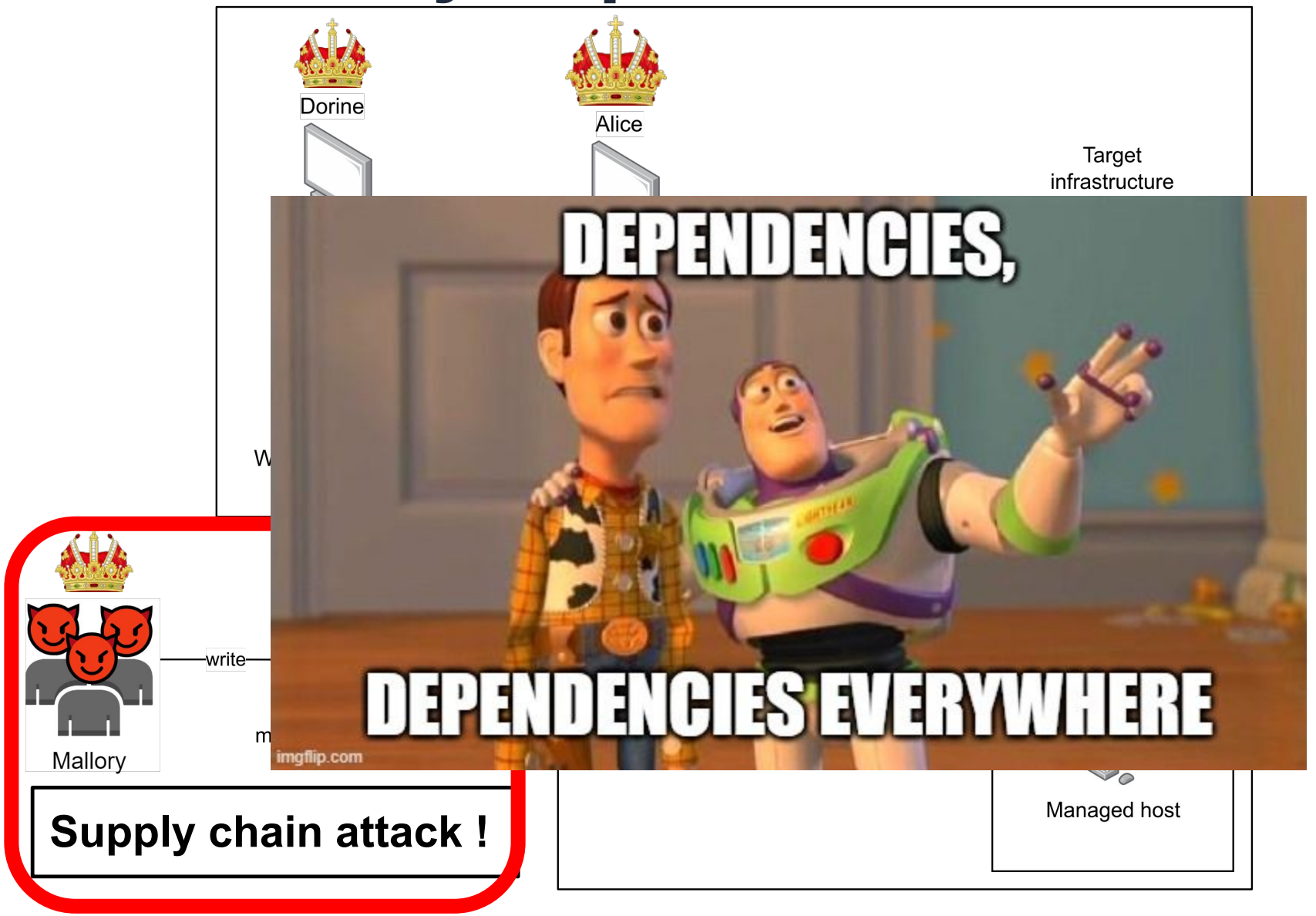
Will you use my tool?

~~Am I trustworthy enough?~~

Is my project trustworthy now?

(Hidden slides for RUMP)

What about my dependencies?



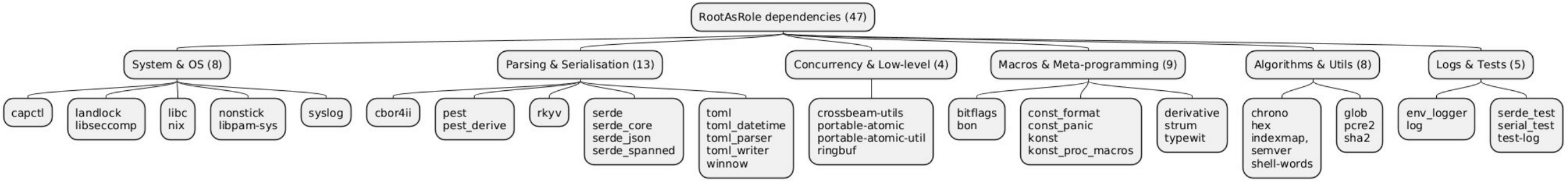
AIRBUS

Are my dependencies trustworthy?

~~47~~ dependencies to address…

~~191~~ with cargo vet

104

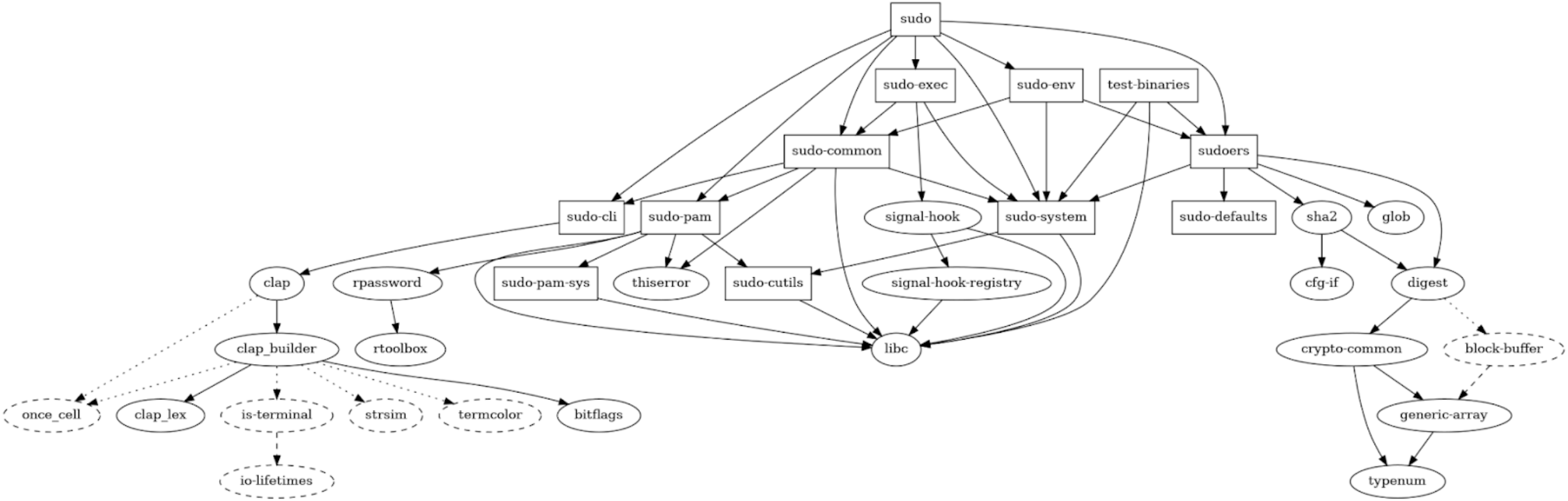


Just 528939 lines of code instead of 2.6 Millions,
What a deal !

AIRBUS

Sudo-rs dependencies:

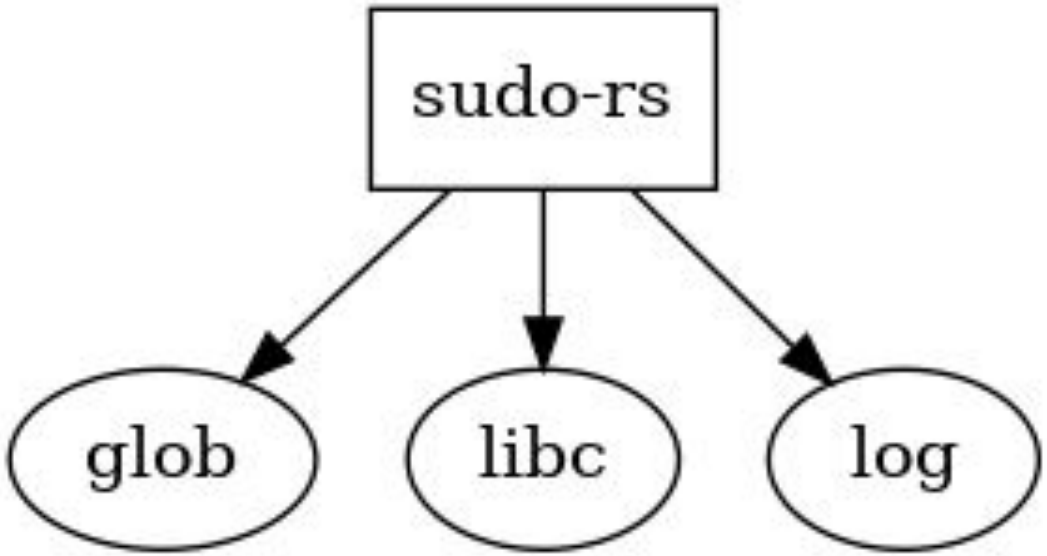
when less is better



AIRBUS

Sudo-rs dependencies:

when less is better



Yes, it's not technically Zero-dependencies… but still

AIRBUS

Wait a minute…

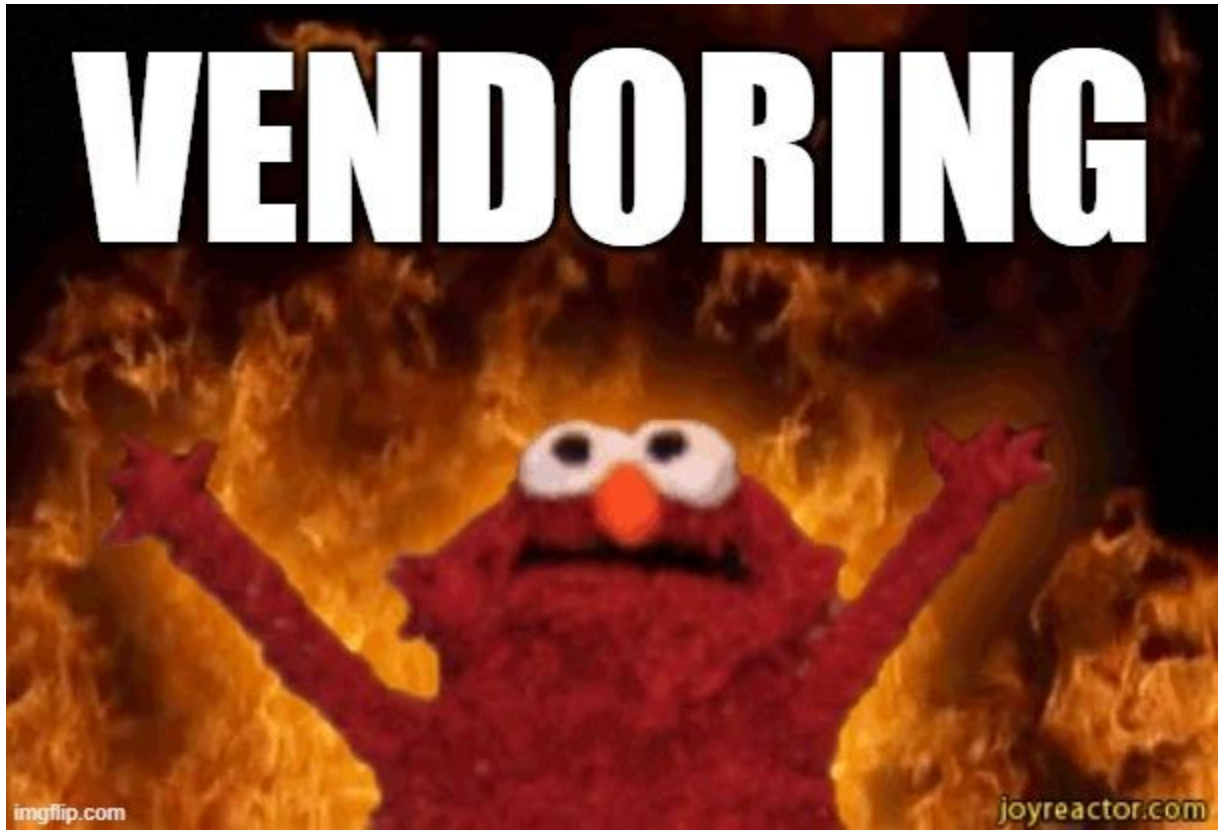
Licensing status: It's a cat→

Where is PAM ??

And clap??

And rpassword??

And signal hooks??



AIRBUS

According to Gemini, Vendoring is a good idea…

1. When the lib is unmaintained → Fork-it
2. For fixing a bug → Fork-it
3. Reducing the amount of dependencies → For bloating?
4. Avoiding an external repository → Fork-it & Artifactory?
5. For guaranteeing security → Fork + security of what ???
6. Modifying the use case for your specifics → Yes

Let’s dive into sudo-rs choices

1. rpassword impl of sudo-rs is adding more security
   - ➢ Why not sharing the security features with a Fork?
2. “PAM because none lib manages both Linux and BSD”
   - ➢ Why not making a library?
3. clap crate don’t fit sudo implementation requirements. OK
4. signal hooks crate design is not matching for sudo-like tool. OK

We have the choice

1. Using many deps to get thousands of lines to review.
   - ➢ Contributing to community dependencies for the security good
   - ➢ Having more optimized and community expertise

2. Vendoring what we need
   - ➢ “For security reasons… we don’t trust the community…”
   - ➢ If someone enhance it on any side, you’ll never be noticed

We could go beyond

- Why about creating a software which is actually a library that anyone could use?
  - Creating a Linux secure command exec crate?
  - Creating a secure signal management crate?
  - Creating a PAM client crate? → tomorrow presentation

→Secure exec crate in v4.0 of RootAsRole!!! 🎉

Available in a few “chore” git commits.

} _enabling a trusted future

PROTECT