- **Regan Robertson** Mod • 15 days ago
  Good Afternoon,
  The video will start on this page at 13:30, and you may have to hit play or unmute. As a reminder this site works best in a Chrome Browser. You can write in comments through this feature to continue the conversation or ask questions.
  - Reply
  - Share ›

Gernot • 15 days ago
Do others also have an echo, a few seconds delayed?
- Reply
- Share ›

Jerry Dussault Gernot • 15 days ago
Not me - no echo.
- Reply
- Share ›

Gernot Jerry Dussault • 15 days ago

sorted it out...

- Reply
- Share ›

Jacob Saina Gernot • 15 days ago

I noticed that if I have nother window open, the video will start in that window as well, with a delay

- 1
- Reply
- Share ›

Nathaniel Husted • 15 days ago

As I'm noticing a theme with Rust & seL4, I just wanted to toss out a resource I found related to verification and Rust in general (vs. seL4 in particular) that may be useful for the community: https://alastairreid.github.... It appears a part of Google is looking into determining how to get Rust-based verification tools into engineers' hands.

- Reply
- Share ›

Arun Thomas Nathaniel Husted • 15 days ago

Thanks, Nathaniel. I read this post (again) while preparing for this talk. It would be awesome if Alistair and Google could put additional resources behind Rust verification.

Reply

Share ›

Jason H Li • 15 days ago

Chris - for verified components, you are following the good works. So am I. Question - do you see some natural way of leveraging some of them? Of course the details are in the details.

Reply

Share ›

Arun Thomas Jason H Li • 15 days ago

Indeed, the details are in the details. As the projects mature, our hope is that we will be able to 1) swap out isolated, security-critical components (e.g, parsers, crypto libraries) for

verified equivalents and 2) adopt new formal verification techniques pioneered by these programs. Time will tell, but we are optimistic.

- Reply
- Share ›

Ihor Kuz • 15 days ago

It would be great to have active Rust support on seL4.

2

Reply

Share ›

Nathaniel Husted Ihor Kuz • 15 days ago

Concur. They seem like such a nice fit.

Reply

Share ›

Jason H Li Nathaniel Husted • 15 days ago

Second that!

- Reply
- Share ›

Noah Evans Ihor Kuz • 15 days ago

What happened to the Lincoln Labs seL4/Rust work?

- Reply
- Share ›

Jason H Li Noah Evans • 15 days ago

Noah - let me ask and find out. Rick was talking about it last year.

- 1
- Reply
- Share ›

Arun Thomas Jason H Li • 15 days ago

The Lincoln work was based on feL4.

- Reply
- Share ›

Arun Thomas Ihor Kuz • 15 days ago

There seems to be a lot of interest in the community. We'd love to work with others on making this happen.

- Reply
- Share ›

Nick Spinale Arun Thomas • 15 days ago

As would we at Arm Research! In addition to engaging in public forums such as https://sel4.discourse.group, I'd be interested in meeting to discuss our efforts with anyone else working on this.

- Reply
- Share ›

Lennart Beringer • 15 days ago

Do you have formal models (in Coq, Isabelle,...) and proven properties about various tag policies?

- Reply
- Share ›

Arun Thomas Lennart Beringer • 15 days ago

Andrew Tolmach's group has been doing working on a verified micropolicy toolchain for the PIPE architecture. I don't have a handy citation for that work just yet, but Andrew's student Sean Anderson gave a related talk at PriSC 2020

- Reply
- Share ›

Olin Sibert • 15 days ago

In what ways do you anticipate changing seL4 or its API to support the PIPE tagging and rule mechanisms?

- 1
- Reply
- Share ›

Arun Thomas Olin Sibert • 15 days ago

In general, our approach has been to minimize changes to software running on the PIPE. So far, we have only made minimal under-the-hood changes to support tagging of seL4 images. It's possible we might want to extend the API in the future as we gain more experience with tagging on seL4, but we would need a compelling reason.

Reply

Share ›

Ihor Kuz • 15 days ago

what micropolicies did you implement for seL4/CAmkES?

1

Reply

Share ›

Arun Thomas Ihor Kuz • 15 days ago

We have only implemented seL4/CAmkES support for a few micropolicies (RWX, stack protection, taint) so far. Some micropolicies were not immediately portable to seL4/CAmkES, since they require a custom clang toolchain we developed. We are working

on addressing this limitation and hope to bring up our full suite of policies -- along with seL4-specific policies -- in the near future.

Reply

Share ›

Alex Glib • 15 days ago

Interesting talk. Can you provide any more info on the methods used to verify the security of each of the separate layers and also the entire integration. Thanks.

Reply

Share ›

Arun Thomas Alex Glib • 15 days ago

Great question. Currently, we are relying on the guarantees provided by each technology (e.g., seL4 proofs). We would like to work toward providing a system-wide guarantee, but that's a long term research challenge.

Reply

Share ›

Regan Robertson Mod • 15 days ago

Please join us for the next session that starts in 2 minutes. You can either go back to agenda to get to the next session or at the bottom of the page there is a next session button.

- Reply
- Share ›

Robbie VanVossen • 15 days ago

Great talk Arun

- Reply
- Share ›

Arun Thomas Robbie VanVossen • 15 days ago

Thanks, Robbie.

- Reply
- Share ›

June Andronick • 15 days ago

How do your HW-enforced SW-defined policies in PIPE interact with seL4 policies (caps)?

Reply

Share ›

—

Arun Thomas June Andronick • 15 days ago

Our current set of policies are mostly orthogonal to seL4 policies. There are some policies, such as compartmentalization, that would require adaption. We will gain a better sense for this as we bring up more micropolicies on seL4.

Reply

Share ›