- **Gernot** • 15 days ago
  Hi, happy to address any questions you have
  - Reply
  - Share ›

Raymond Richards Gernot • 15 days ago
Gernot, good to hear from you, I hope all is well.

Reply

Share ›

Jason H Li • 15 days ago
Congrats on this new milestone for RISC-V.

3

Reply

Share ›

Jason H Li • 15 days ago

Do you have a paper describing the proof of FC on RISC-V? If so, we would love to read it.

- Reply
- Share ›

Gernot Jason H Li • 15 days ago

No paper, it's the same approach as taken for Arm and x86. Although, with 3 architectures now it's probably time for an experience paper

- 2
- Reply
- Share ›

Jason H Li Gernot • 15 days ago

Lessons learned would be great for the community!

- Reply
- Share ›

Jerry Dussault • 15 days ago

I'm having brief audio/video drop-outs. Anyone else?

- 4
- •
- Reply
- •
- Share ›

Jacob Saina Jerry Dussault • 15 days ago

yes

- •
- Reply
- •
- Share ›

Todd Carpenter Jacob Saina • 15 days ago

Yes

- •
- Reply
- •
- Share ›

Jason H Li Jacob Saina • 15 days ago

same. off and on.

- Reply
- Share ›

Kyle Tillotson Jerry Dussault • 15 days ago

Same here

- Reply
- Share ›

Renato Levy Jerry Dussault • 15 days ago

i already contacted the hub management. They are looking onto it

- Reply
- Share ›

Regan Robertson Mod Jerry Dussault • 15 days ago

Please refresh your page to watch the video as we took the presentation off of live streaming.

- Reply
- Share ›

Regan Robertson Mod • 15 days ago

We had some buffering issues on our end, so we decided to re-load the recorded talk.

- Reply
- Share ›

Regan Robertson Mod • 15 days ago

Please refresh your page for the recorded video and fast forward to the spot we left off at.

- Reply
- Share ›

Raymond Richards • 15 days ago

How are proofs over abstract model architecture dependent?

1

Reply

Share ›

Jason H Li Raymond Richards • 15 days ago

Ray - my understanding is that an abstracted model is a refinement of another model for FC. This does not involve a particular architecture, yet.

But of course Gernot has the exact answer.

Reply

Share ›

Ihor Kuz Jason H Li • 15 days ago

Also Matt has a talk about the proofs tomorrow: https://www.sel4summit.com/...
If you don't get the answer today, you can ask Matt then.

Reply

- Share ›

Gernot Jason H Li • 15 days ago

seL4 isn't a hardware abstraction, hence the spec depends (somewhat) on the underlying ISA

- Reply
- Share ›

Jason H Li Gernot • 15 days ago

Understand, and thanks for the clarification.

- Reply
- Share ›

Ihor Kuz Raymond Richards • 15 days ago

The kernel has architecture dependent datastructures/abstractions, which means the API is actually different for the different architectures.

Reply

Share ›

Jason H Li Ihor Kuz • 15 days ago

Yep this makes total sense.

Reply

Share ›

Matthew Brecknell Raymond Richards • 14 days ago

Hi Ray, I tried to reply to this several times, but Disqus keeps flagging my post as spam, even though I'm logged in and successfully identified the fire hydrants. Perhaps it was too long, so I'll break it up into a series...

Reply

Share ›

Matthew Brecknell Matthew Brecknell • 14 days ago

1/4: I guess your question is about Gernot's comment in the talk that the security proofs currently only work for the 32-bit ARMv7 port (without hypervisor). As Ihor and Gernot mentioned in this chat, the specification contains architecture-dependent aspects, so the security proofs need to deal with those. We have no reason to believe that other ports wouldn't satisfy the security properties, but we've not yet worked through the proofs.

Reply

Share ›

Matthew Brecknell Matthew Brecknell • 14 days ago

2/4: Why does the abstract specification contain architecture-dependent aspects? The "abstract" in our "abstract specification" really only means that it's optimised for clarity and formal reasoning, as opposed to the implementation which is optimised for efficiency and executability. We actually intend the abstract spec to precisely capture all of the kernel behaviour on which user space might want to depend. As a microkernel, seL4 tries to delegate as much as possible to userspace, and the only way to do that without compromising performance or imposing policy is to expose a lot of architecture-dependent detail in the API. And so the abstract spec necessarily contains that architecture-dependent detail.

Reply

Share ›

Matthew Brecknell Matthew Brecknell • 14 days ago

3/4: The most significant example of architecture-dependent detail in the API is virtual address space management. The seL4 API gives user space (almost) complete control of virtual memory, with capabilities for pages and page tables. Architecture-dependent details exposed here include the number of levels in the page table hierarchy, the number of address bits decoded by each level, and the attributes and rights that can be set on page table entries. ARMv7's special treatment of large pages and super sections is also exposed.

Reply

Share ›

Matthew Brecknell Matthew Brecknell • 14 days ago • edited

4/4: Tangentially related: One of our current biggest pure engineering challenges is what we call the "matrix" problem. The kernel supports multiple architectures and platform configurations, and has various build configuration options. This creates a matrix of configurations for which we would like to have proofs. However, we currently only have proofs for a few specific configurations, and some of the proofs (e.g. security) only work on a subset of those configuration. It's still quite a lot of effort to add proofs for a new configuration, and also to maintain them. We do have ideas for improving the way we handle architecture-dependent aspects, and I think these could make it much easier to achieve broader coverage of the matrix, and also to get more of the proofs working on more configurations. Since virtual address space management domainates the architecture-dependent parts of the specification, I think the first step is a more abstract treatment of page table mappings across all configurations. The specification for the RISC-V seL4 port already takes some steps in this direction, but we need to find time and resources (i.e. funding!) to keep pursuing that...

Reply

Share ›

Jason H Li • 15 days ago

To connect with binary for RISC-V, do you have something like Myreen's decompiler for ARM? Or your group has to come up with something similar by yourselves?

Reply

Share ›

Ihor Kuz Jason H Li • 15 days ago

Yes (the former).

Reply

Share ›

Gernot Jason H Li • 15 days ago

We use Myreen's decompiler, long-standing collaboration with Magnus

Reply

Share ›

Lennart Beringer • 15 days ago

Is there a pointer to the performance evaluation paper?

Reply

Share ›

Gernot Lennart Beringer • 15 days ago

which performance eval are you referring to?

Reply

Share ›

Lennart Beringer Gernot • 15 days ago

I meant the evaluation by some other OS group you referred to early in the slide and hoped there was a paper for it. But maybe the citations given 2mins later in the bottom half of the slide contain all the info...

Reply

- Share ›

Gernot Lennart Beringer • 15 days ago

yes, they are the two papers, the numbers are taken from there

Reply

Share ›

Olin Sibert • 15 days ago

To what extent do the cycle counts on slide 9 incorporate any effects of cache turbulence across context switches?

Reply

Share ›

Gernot Olin Sibert • 15 days ago

these are best-case times, hot cache

- Reply
- Share ›

Todd Carpenter • 15 days ago

Wow, Gernot, you really don't like the domain scheduler, do you. Too bad it's so easy to use, and it just works!

- 3
- Reply
- Share ›

Jacob Saina • 15 days ago

What does MCS stand for?

- Reply
- Share ›

Ihor Kuz Jacob Saina • 15 days ago

mixed criticality systems

- Reply
- Share ›

Jason H Li Jacob Saina • 15 days ago

mixed criticality

- Reply
- Share ›

Gernot Jacob Saina • 15 days ago

mixed-criticality systems (support)

- 1
- Reply
- Share ›

Fabrizio Bertocci Jacob Saina • 15 days ago

https://docs.sel4.systems/T...

Reply

Share ›

Noah Evans • 15 days ago

Gernot have you ever looked at running seL4 on extremely resource constrained systems (e.g. cacheless, prepaged)? Would that make the real time proofs easier? I know the eChronos is likely the correct choice here, but I'm curious about the thought experiment.

Reply

Share ›

Gernot Noah Evans • 15 days ago

less microarchitectural state simplifies things, of course. Eliminating timing channels in microcontrollers is likely straightforward. But I'm looking for a principled solution that works for high-end processors

Reply

Share ›

Noah Evans Gernot • 15 days ago

Understood, have you seen any customers interested in these sorts of resource constrained systems?

- Reply
- Share ›

Todd Carpenter • 15 days ago

That was a great summary of the progress this year! I particularly appreciate the update on temporal partitioning. It seems like it was a very productive year. I look forward to using MCS as soon as the verification team lets it out of their grip.

- 2
- Reply
- Share ›

Jason H Li Todd Carpenter • 15 days ago

Same here. Look forward to using MCS!

- Reply
- Share ›

Gernot Jason H Li • 15 days ago

MCS verification is a few months away, but that shouldn't stop you using it. Unless you're working on something you want to deploy in Q1

Reply

Share ›

Renato Levy • 15 days ago

Since all architectures are k-associative, and if you are using each association for a different domain, would that not limit the total number of domains?

1

Reply

Share ›

Gernot Renato Levy • 15 days ago

We don't partition by associativity (that needs HW support, such as Intel CAT), we partition by colour. But yes, there's a limit to the number of colours. But it only needs to be sufficient for the working set of security domains

Reply

Share ›

Carl Nerup • 15 days ago

Brilliant. We are on the right tool. Thanks to you and everyone at the Foundation.

Reply

Share ›

Renato Levy • 15 days ago

this is the first time that i see timing channels properly addressed. Great work! very impressive

Reply

Share ›

Gernot Renato Levy • 15 days ago

I like tackling problems everyone else puts into the too-hard basket. That's how seL4 was born ;-)

- 1
- Reply
- Share ›

Olin Sibert • 15 days ago

If one is stuck with an architecture (e.g., ARM64) that doesn't have great support for clearing microarchitectural state, how plausible do you think it is to limit access to clock sources (both internal and external) as a partial mitigation to timing channels? Closely related, is it plausible to imagine mathematically modeling potential transmission bandwidth in a meaningful way on such platforms, such that the proof would prove a numerical answer rather a simple yes/no?

- Reply
- Share ›

Renato Levy Olin Sibert • 15 days ago

Do you want to know if your leak is dripping or pouring?

- Reply
- Share ›

Olin Sibert Renato Levy • 15 days ago

Yes. Ultimately, security is about risk *management*, not just about risk elimination. So it's completely legitimate to imagine trading performance for transmission bandwidth, but only if you can have some plasuible quantification of risk.

- Reply
- Share ›

Gernot Olin Sibert • 15 days ago

completely virtualising time is virtually (excuse pun) impossible in practice, except in very restricted setups.

Quantifying leakage is possible, but remember, even 1 bit/s leaks an SSL key in a few minutes

- Reply
- Share ›

Olin Sibert Gernot • 15 days ago

Quite so. This was one of the places where the Digital SVS project did fundamental work but eventually foundered. But even partial virtualization provides a means to

reduce leakage and make trades. Back in the 1980s, though, there was very little mathematics behind it, and I'm hoping that the intervening time might have improved that situation.

Reply

Share ›

Hugo Vincent Olin Sibert • 15 days ago

Channel capacity is a well-understood metric for quantifying these leaks, yes. It is useful as it allows you to mitigate at higher levels, e.g. rotate keys before enough bits have leaked.

Reply

Share ›

Hugo Vincent • 15 days ago

Hi Gernot, great talk! Do you have any data (even prelim) on performance impact of time protection? If I understood correctly, you pad (spin loop up to WCET), flush lots of close-to-core state and partition (i.e. reduce effective size of) L2/L2 etc, all of this must have a substantial cost.

Reply

Share ›

Gernot Hugo Vincent • 15 days ago

Yes, there's performance data in the 2019 EuroSys papers. tl;dr it's negligible

- Reply
- Share ›

—

Gernot • 15 days ago

Need to switch to the pannel now ;-)

- Reply
- Share ›