Verified seL4 on Secure RISC-V Processors

... and Other News in seL4 Land

Gernot Heiser | gernot.heiser@data61.csiro.au | @GernotHeiser
• LCA, Gold Coast, QLD, 2020-01-15

https://trustworthy.systems
What is seL4?
A 30-Year Dream

1. Introduction

Early attempts to make operating systems secure were fraught with difficulties. As these efforts failed, it became clear that stringent design principles were needed to ensure feasibility. A more recent approach is to employ a combination of techniques that can be integrated into the kernel's design and implementation. This approach is based on the observation that secure systems are often secure in the same sense that an impenetrable fortress is secure - not because it cannot be breached, but because the breach would be detected in time to prevent access.

Our research seeks to develop means by which an operating system can be shown data secure, meaning that direct access to data must be possible only if the recorded protection policy permits it. The two major components

Communications of the ACM

February 1980

Volume 23

Number 2

SEL4
seL4: The Dream Come True!

- The world’s **first** operating-system kernel with **provable** security enforcement
- World’s most advanced mixed-criticality OS
- The world’s **only** protected-mode OS with complete, sound timeliness analysis
- The world’s **fastest** microkernel, designed for **real-world** use
- Open Source
L4: 25 Years High-Performance Microkernels

seL4: The latest member of the L4 microkernel family

API Inheritance

Code Inheritance

L3→L4

L4/MIPS

L4/Alpha

"X"

Hazelnut

Pistachio

L4-embed.

OKL4-µKernel

Qualcomm modem chips

OKL4-Microvisor

Codezero

seL4

GMD/IBM/Karlsruhe

UNSW/NICTA/Data61

OK Labs

Dresden

Other (commercial)

Fiasco

Fiasco.OC

Nova

P4 → PikeOS

seL4: The latest member of the L4 microkernel family

API Inheritance

Code Inheritance

L3→L4

L4/MIPS

L4/Alpha

"X"

Hazelnut

Pistachio

L4-embed.

OKL4-µKernel

Qualcomm modem chips

OKL4-Microvisor

Codezero

seL4

GMD/IBM/Karlsruhe

UNSW/NICTA/Data61

OK Labs

Dresden

Other (commercial)

Fiasco

Fiasco.OC

Nova

P4 → PikeOS
A Microkernel is not an OS

Device drivers, file systems, crypto, power management, virtual-machine monitor are all usermode processes

Microkernel = context-switching engine

Strong Isolation

Controlled Communication
Core Mechanism: Object Capability

Capability = Access Token: Prima-facie evidence of privilege

- Obj reference
- Access rights

Eg. read, write, send, execute…

Object

Eg. thread, address space

Capabilities provide:
- Fine-grained access control
- Reasoning about information flow
Microkernel: seL4 vs Linux Licensing

Valuable IP

Your code (applications)
any...

TS system services:
- libs, drivers, file systems,
- NW stacks, tools, ...
BSD

Your system services
any...

Platform port:
- timer
- serial
- HW init

Boiler plate

ts kernel source
GPL v2

kernel.org source
- core kernel
- device drivers
- file systems
- NW stacks
- ...
GPL v2

Your code (applications)
any...

Platform port:
- device drivers
- HW init

GPL v2

Boiler plate

Details: https://microkerneldude.wordpress.com
Military-Strength Security

DARPA HACMS: Retrofit existing system!

Unmanned Little Bird (ULB)

Secure Comms Dongle

Autonomous trucks

Cross-Domain Desktop Compositor
Verification
World’s Most Secure OS: Arm v7

Confidentiality
Integrity
Availability

Abstract Model

Translation validation:
Binary retains C-code semantics

C Implementation

Sound worst-case execution time bound

Binary code

Model enforces security

Functional correctness:
C code only behaves as specified

Limitations (work in progress):
• Kernel initialisation not yet verified
• MMU & caches modelled abstractly
• Timing channels not ruled out

C Code only behaves as specified
Background: HENSOLD Cyber

Munich-based startup
- Secure RISC-V processor
- Based on open-source Ariane
- Supply chain secured through logic encryption
- Secure OS based on seL4
- Targets defence, industrial control, critint, automotive

Disclosure: I have an interest in HENSOLDT Cyber
## Performance on RV64

**Message-passing round-trip latency in cycles**

<table>
<thead>
<tr>
<th>Arch</th>
<th>x86 32b</th>
<th>x86 64b</th>
<th>Arm 32b</th>
<th>Arm 64b</th>
<th>RISC-V 64b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra address space</td>
<td>427</td>
<td>565</td>
<td>625</td>
<td>752</td>
<td>690</td>
</tr>
<tr>
<td>Inter address space</td>
<td>752</td>
<td>1041</td>
<td>625</td>
<td>752</td>
<td>1006</td>
</tr>
</tbody>
</table>

*Not yet fully optimised!*

- **Spectre-workaround disabled** (else much more expensive)
- **No ASIDS on HiFive Unleashed**, else inter-AS would be same as intra-AS
- Hypervisor extensions (draft spec 0.5) supported in branch
Verification: RISC-V Status

- Confidentiality
- Integrity
- Availability

Proof

Abstract Model

Proof

Functional correctness: RISC-V due Q1’20

Translation validation: RISC-V due Q2’20

Proof

C Implementation

Proof

Sound WCET bound RISC-V in progress

Proof

Binary code
Experience with RISC-V Architecture

- Kernel port straightforward:
  - simple and clean RISC architecture
- Verification benefitted from cleanness
  - … but some challenges from less typing in page tables
- Hypervisor (draft) extensions even simpler
- M (machine) mode makes firmware explicit
  - configures HW, delegates to S (supervisor) mode
  - emulates features not implemented in HW
  - should be verified
- Extensibility of ISA could be a concern
  - could undermine portability
- Formal ISA spec is great!
LCA’18 Refresher: Time Capabilities

Classical thread attributes
- Priority
- Time slice

New thread attributes
- Priority
- Scheduling context capability

Scheduling context object
- T: period
- C: budget (≤ T)

Limits CPU access!

Not runnable
if null

Capability for time

Enables reasoning about
time and temporal isolation
for mixed-criticality systems

C = 2
T = 3

C = 250
T = 1000
Time Caps (MCS) Kernel Verification

- **Mainline Arm v7**
  - Spec
  - **Proof**
  - **C**
  - Binary

- **MCS Arm v7**
  - Spec
  - **Proof**
  - **C**
  - Binary

- **MCS RISC-V**
  - Spec
  - **Proof**
  - **C**
  - Binary

- **Mainline RISC-V**
  - Spec
  - **Proof**
  - **C**
  - Binary

**New Mainline**

- Q1'20
- Merge
- Q2'20
- Merge
- Q1'20
- Merge
- Q2'20

**Proof**

- Q1'20
- Q2'20
- Q2'20
- Q1'20
Research: Time Protection
Threats

Speculation
An “unknown unknown” until recently

A “known unknown” for decades

Microarchitectural Timing Channel
Cause: Competition for HW Resources

High

Low

Affect execution speed

- Inter-process interference
- Competing access to micro-architectural features
- Hidden by the HW-SW contract!
Sharing: Stateful Hardware

HW is capacity-limited
• Interference during concurrent access
• time-shared access
• Collisions reveal addresses
• Usable as side channel

High

Low

Cache

Any state-holding microarchitectural feature:
• cache, branch predictor, pre-fetcher state machine
Time Protection: Prevent Interference

Affect execution speed

Interference results from sharing ⇒ Partition hardware:
- spatially
- temporally (time shared)
Time Protection: Partition Hardware

**Temporally partition**

- **Flush**
  - Need both!
  - Cannot spatially partition on-core caches (L1, TLB, branch predictor, pre-fetchers)
    - virtually-indexed
    - OS cannot control

**Spatially partition**

- Flushing useless for concurrent access
  - HW threads
  - cores
- Partitions get frames of disjoint colours
- seL4: userland supplies kernel memory ⇒ colouring userland colours dynamic kernel memory
- Per-partition kernel image to colour kernel

[Ge et al. EuroSys’19]
Temporal Partitioning: Flush on Switch

1. \( T_0 = \text{current\_time}() \)
2. Switch user context
3. Flush on-core state
4. Touch all shared data needed for return
5. while \( (T_0 + \text{WCET} < \text{current\_time}()) \) ;
6. Reprogram timer
7. return

Latency depends on prior execution!

Time padding to Remove dependency

Ensure deterministic execution

Must remove any history dependence!
Challenge: Broken Hardware

- Systematic study of COTS hardware (Intel and Arm) [Ge et al, APSys’18]:
  - contemporary processors hold state that cannot be reset
Way Out: New HW-SW Contract!

ISA is purely functional contract, abstracts too much away

New contract (augmented ISA):
All shared HW resources must be spatially or temporally partitionable by OS
[Ge et al, APSys’18]

RISC-V to the rescue:
Strong commitment to making it happen!
Community/Ecosystem
Experience with RISC-V Foundation

Security Standing Committee
- Invited me on
- Very receptive and supportive
- Committed to making RISC-V “most secure architecture”
- Facilitated engagement with Privspec TC (now Standing Committee)

Privileged Spec Tech Committee
- Hypervisor-extension feedback well received
  - Easy engagement
  - Constructive proposal from TC chair addressing our issues
- Time-protection slow to get traction
  - Now good engagement, hopefully progress soon

- Open but skeptical
- They need to manage conflicting ideas
- Keen to get “most secure arch” recognition
We Are Creating the seL4 Foundation!

Aims:

• Provide a neutral entity for coordinating & enhancing seL4 ecosystem
• Grow adoption of seL4
• Improve (organisational and individual) community participation & cooperation
  • Developers
  • Adopters
• Develop / standardise seL4 system
  • kernel & proofs
  • libraries, services, tools
• Protect and promote the seL4 brand
  • prevent reputational damage from using modified seL4 (verification invalidated)
• Provide platform for pooling funds for critical “big-ticket” items (verification)
Foundation Structure

- seL4 Foundation
- seL4 Board
- seL4 Fund Charter
- seL4 Directed Fund $$
- seL4 Series LLC
- seL4 Technical Charter
- LF Projects LLC
- seL4 TM
- https://sel4.systems
- Technical Project
- Contributor
Membership (Subject to Minor Change)

- **Trustworthy Systems**: 3 directors
- **Premium Members US$ 100k/a**: 1 director each
- **Members US$ 3–30k/a**: 1 director

**Chair ex officio**

**Technical Steering Committee**

**Technical Leader(s)**

**Committer**

Initial Board:
- June Andronick, TS
- Gernot Heiser, TS
- Gerwin Klein, TS
- John Launchbury, Galois (ex DARPA)
- Sascha Kegreiß, HENSOLDT Cyber
- Daniel Potts, Ghost Locomotion

Note: members must be financial members of Linux Foundation!
Community Engagement

Trustworthy Systems Team
- Maintain/extend
- Evolve
- Provide & maintain

Core userland
- Proofs
- Code

Other userland
- Adopt/extend/maintain/innovate!
- Contribute, adopt?
- Platform ports

Community

LCA | Gold Coast | Jan'20
Foundation Status

- Legal docs (fund charter & technical charter) submitted to Linux Foundation
  - just received their feedback
- Trademark ready for transfer to Foundation
- Initial board appointed
- Interim web site shows structure and “Principles” document
  - legal docs will be there once approved by LF
- Hopefully days away from being able to set up members
  - Mail foundation@sel4.systems if you’re interested in joining!

https://sel4.systems/Foundation
THANK YOU

Gernot Heiser | gernot.heiser@data61.csiro.au | @GernotHeiser
• LCA'20, Gold Coast, QLD, 2020-01-15

https://trustworthy.systems