Above and Beyond: seL4 Noninterference and Binary Verification

Toby Murray and Thomas Sewell *

NICTA and the University of New South Wales, Australia
firstname.lastname@nicta.com.au

*joint work with Matthew Brassil, Timothy Bourke, Peter Gammie, Xin Gao, Gerwin Klein, Corey Lewis, Daniel Matichuk and Magnus O. Myreen

In 2009, the L4.verified project completed the world's first verification of *functional correctness* for a general-purpose OS kernel [2], seL4. Functional correctness here was embodied by a formal theorem of *refinement*, which stated that the behaviour of the C code that implemented the kernel accorded with an *abstract specification* of how the kernel was meant to function.

A cynic would say that this result proves only that the kernel has no bugs that are not present in its abstract specification or the C compiler. Specifically, while seL4 was designed for security, the functional correctness proof did not rule out the abstract specification specifying insecure behaviour. Further, there was no guarantee that the compiler used to compile the kernel's C code would produce a binary whose behaviour matched its formal C semantics from the functional correctness proof: the proof did nothing to rule out compiler bugs.

Therefore, two obvious questions remained. Firstly, how do we know that the abstract specification correctly specifies the kernel we want? Secondly, how do we know that the compiler will honour the formal C semantics of seL4?

We present new results on both of these fronts. First, we discuss the recent proof of the classical security property of *noninterference* for seL4 over its abstract specification [3]. By virtue of the functional correctness proof, we obtain this result automatically for the kernel's implementation—a world first for a general purpose kernel. Coupled with an earlier proof of integrity for seL4 [5], we now *know* that it is secure: seL4 provably enforces confidentiality and integrity.

Second, we discuss recent work on applying *translation validation* to seL4, to formally verify that the binary produced by the compiler adheres to the formal C semantics used in the functional correctness proof [4]. Here, we interpret the binary against a formal model of the ARM ISA developed by Fox et al. [1] to give it a formal semantics, which is then matched-up against the formal semantics of the kernel's C code, and the match automatically proved by SMT solving.

Combined, these results give us security theorems that apply to the kernel binary; neither the abstract specification nor the compiler need be trusted anymore. seL4 has now become the world's most deeply verified general-purpose kernel, and so serves as an ideal base for the construction of trustworthy systems. Importantly, system builders can make use of the high level security theorems to obtain system-wide guarantees without needing to trust their kernel or compiler. No other system provides this level of assurance.

Acknowledgements NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. Part of the research reported in this document was performed in connection with Contract Number DAAB W15P7T-12-C-A105 with the US Army CECOM LCMC Command. The views and conclusions contained in this document are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the US Army CECOM LCMC Command or the US Government unless so designated by other authorized documents. Citation of manufacturers or trade names does not constitute an official endorsement or approval of the use thereof. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] A. Fox and M. Myreen. A trustworthy monadic formalization of the ARMv7 instruction set architecture. In *1st ITP*, volume 6172 of *LNCS*, pages 243–258. Springer-Verlag, Jul 2010.

[2] G. Klein, K. Elphinstone, G. Heiser, J. Andronick, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Winwood. seL4: Formal verification of an OS kernel. In *22nd SOSP*, pages 207–220, Big Sky, MT, USA, Oct 2009. ACM.

[3] T. Murray, D. Matichuk, M. Brassil, P. Gammie, T. Bourke, S. Seefried, C. Lewis, X. Gao, and G. Klein. seL4: from general purpose to a proof of information flow enforcement. In *IEEE Symp. Security & Privacy*, Oakland, CA, May 2013.

[4] T. Sewell, M. Myreen, and G. Klein. Translation validation for a verified OS kernel. In *PLDI 2013*. ACM, 2013. To appear.

[5] T. Sewell, S. Winwood, P. Gammie, T. Murray, J. Andronick, and G. Klein. seL4 enforces integrity. In *2nd ITP*, volume 6898 of *LNCS*, pages 325–340. Springer-Verlag, Aug 2011.