Above and Beyond: seL4 Noninterference and Binary Verification (PT. 2)

Submitted by TimTh on Tue, 05/28/2013 - 1:17pm. Contributors: Thomas SewellToby Murray

ABSTRACT:

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 behavior 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 behavior. Further, there was no guarantee that the compiler used to compile the kernel’s C code would produce a binary whose behavior 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 honor 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.


This is a join talk given by two speakers, Toby Murray and Thomas Sewell, of NICTA and UNSW. The slides attached were presented by Thomas Sewell. Toby Murray’s slide deck can be found here.

**BIO:**

**Toby Murray** is a researcher with NICTA, whose interests focus on how to apply formal methods and verification to build more secure systems. He recently led the successful verification of information flow security for the seL4 microkernel, the result of which was the world’s first proof of confidentiality that applied to the implementation of a general-purpose OS kernel. He previously completed a DPhil (PhD) at Oxford under the supervision of Gavin Lowe, researching the application of the process algebra CSP to reason about the security of object-capability patterns, and has also worked for the Australian Defence Science and Technology Organisation (DSTO), researching and building security architectures for pervasive systems.

**Thomas Sewell** is a researcher with NICTA, whose interests focus on program verification, programming languages and operating systems. Specific interests include language semantics, refinement calculi, interactive theorem provers and proof automation approaches.

Thomas is working on extensions to the verification of the seL4 microkernel. He was a core team member of the L4.verified project that proved the functional correctness of the kernel. Since the completion of that project he has extended the functional correctness proof to include the fastpath and worked on an access control model for seL4. He is now working on eliminating the compiler correctness assumption from the seL4 proof by establishing that the compilation was faithful to the assumed semantics of the C code.

Thomas has a combined Bachelor of Engineering (Software Engineering) and Bachelor of Science (Pure Mathematics) degree from UNSW.

<table>
<thead>
<tr>
<th>Thomas Sewell</th>
<th>Toby Murray</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>License:</strong></td>
<td>Creative Commons 2.5</td>
</tr>
</tbody>
</table>

Other available formats:

**Above and Beyond: seL4 Noninterference and Binary Verification (PT. 2)**

Switch to normal viewer
Switch to experimental viewer

[Validation and Verification](#) NICTA UNSW Academia Presentation HCSS'13 HW/SW (HCSS'13)