Reto
Buerki

Adrian-Ken
Rueegsegger

Graduate Candidates	Reto Buerki , Adrian-Ken Rueegsegger
Examiner	Prof. Dr. Andreas Steffen
Co-Examiner	Prof. Dr. Endre Bangerter
Subject Area	Software and Systems

Muen

An x86/64 Separation Kernel for High Assurance

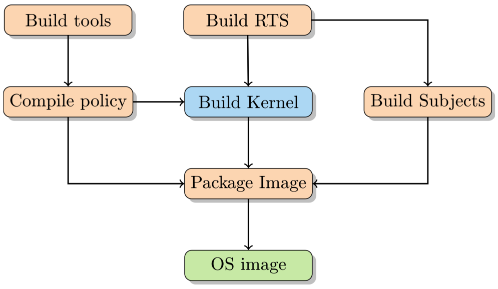

Figure 1: Build process of a system image

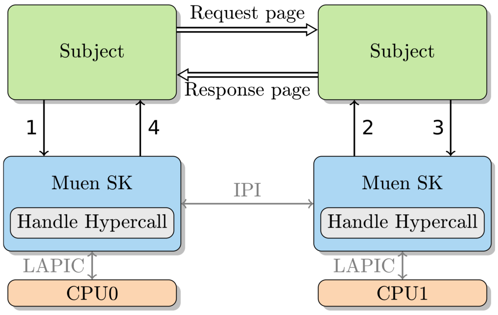

Figure 2: Handling of inter-core events on multi-processor systems

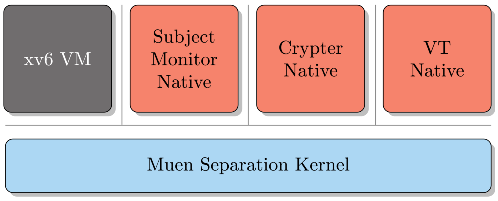

Figure 3: Subjects isolated by the Muen Separation Kernel

Introduction: As computer systems are entrusted with more and more sensitive and personal information, the need to effectively control access to the data becomes increasingly important. Recent revelations about very sophisticated and targeted attacks as well as broad, nation-wide surveillance programs seriously call the effectiveness of currently deployed security systems in question. A common defense strategy is to compartmentalize information and its processing. An example would be the usage of a dedicated computer for Internet banking that is only connected to the Internet when needed. However, this approach does not scale well, since this would necessitate having a separate device for each task that should be performed in some form of isolation.

Approach/Technologies: A separation kernel (SK) is a specialized microkernel which provides an execution environment for components that can only communicate according to a given policy and are otherwise isolated from each other. This isolation also includes the limitation of potential side- and covert channels. Recent addition of advanced hardware virtualization support for the Intel x86 architecture has the potential of greatly simplifying the implementation of an SK. Using hardware virtualization features for component separation and leveraging Intel's latest processor features enables the implementation of a small kernel suitable for formal verification. Using SPARK as the programming language greatly increases the confidence in the implementation since it eliminates complete categories of common programming errors, e.g. buffer overflows. Making the source code and technical documentation publicly available enables third-party review.

Result: The main results of this work are the separation kernel design and prototype as well as the accompanying proof artifacts. To our knowledge it is the first publicly and freely available separation kernel for the Intel x86/IA-32e architecture. Use of the SPARK language and tools for the realization of the Muen kernel provides the base for a high assurance implementation. Full absence of runtime errors and some additional properties have been proven. Focus on use of the latest Intel hardware virtualization features and emphasis on a simple design have resulted in a small code size. It is a sound basis for further formal verification. An example system modeled after a realistic component-based use-case has been implemented to demonstrate the viability of the design and the usability of the kernel prototype.