Code-centered kernel compartmentalization in CheriBSD

Konrad Witaszczyk, Department of Computer Science and Technology, University of Cambridge, March 2025

The CHERI-extended CheriBSD kernel based on the FreeBSD kernel, as other monolithic kernels, consists of millions of trusted lines of code that are compiled into the kernel binary or separate kernel modules. The kernel binary itself includes code from over 1,700 ELF object files linked together. In this research, I explore compartmentalization of the CheriBSD kernel that focuses on its code (Figure 1) and aims to split the kernel code into compartments that can call functions of other compartments only if a system-defined policy allows such a call.

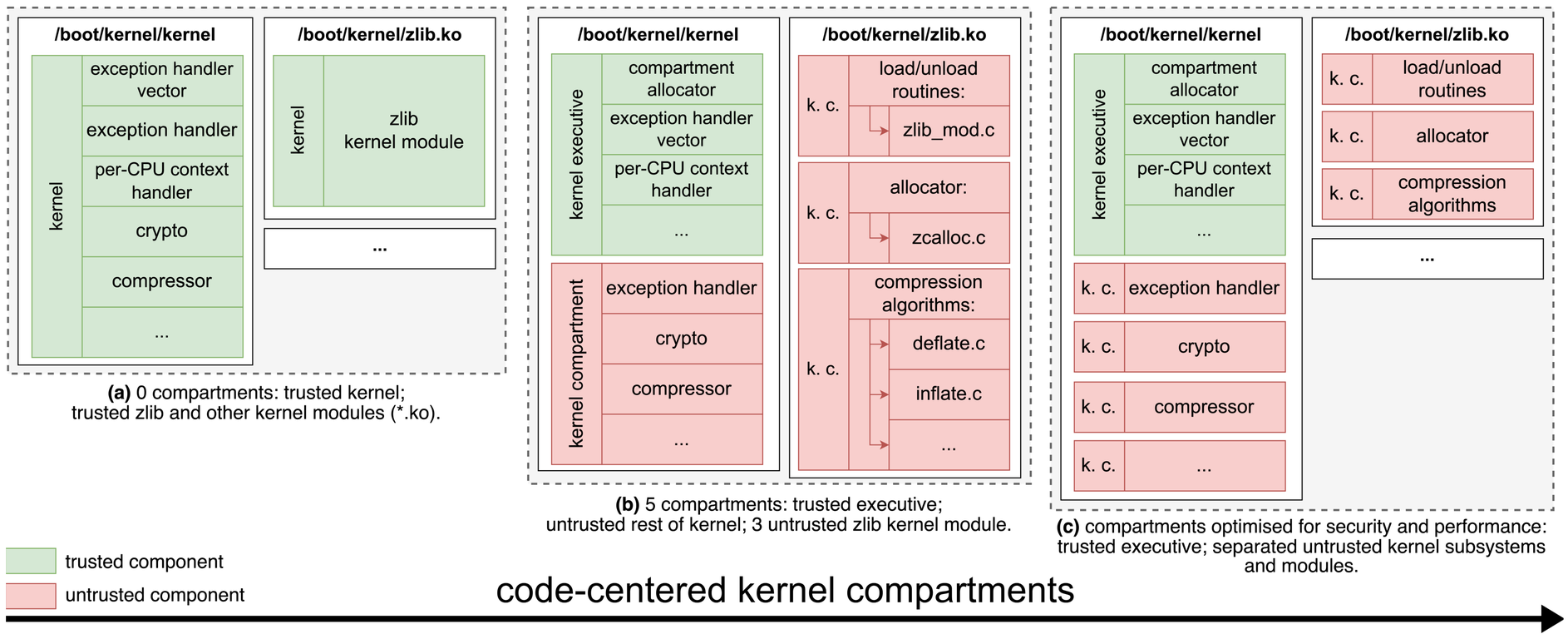


Figure 1. Incremental code-centered compartmentalization approach in a monolithic kernel. Case b presents the current state.

Static and dynamic linking for compartmentalization

The kernel and kernel modules are statically linked into binaries that consist of object files grouped into compartments. When relocating their symbols at run time, the dynamic kernel linker wraps the symbols with trampolines that implement compartment switching.

Calls between compartments

The trampolines (Figure 2) use architectural features (e.g., the Executive and Restricted modes in Arm Morello) to protect capability registers from being manipulated by an untrusted compartment. Those features are specific to a CHERI-extended architecture and are co-designed with software.

Optimal compartment boundaries

The security and performance characteristics of the system can significantly differ depending on the choice of the Trusted Computing Base and untrusted compartment boundaries. The search for the optimal compartmentalization policy requires extensive experiments and engineering.

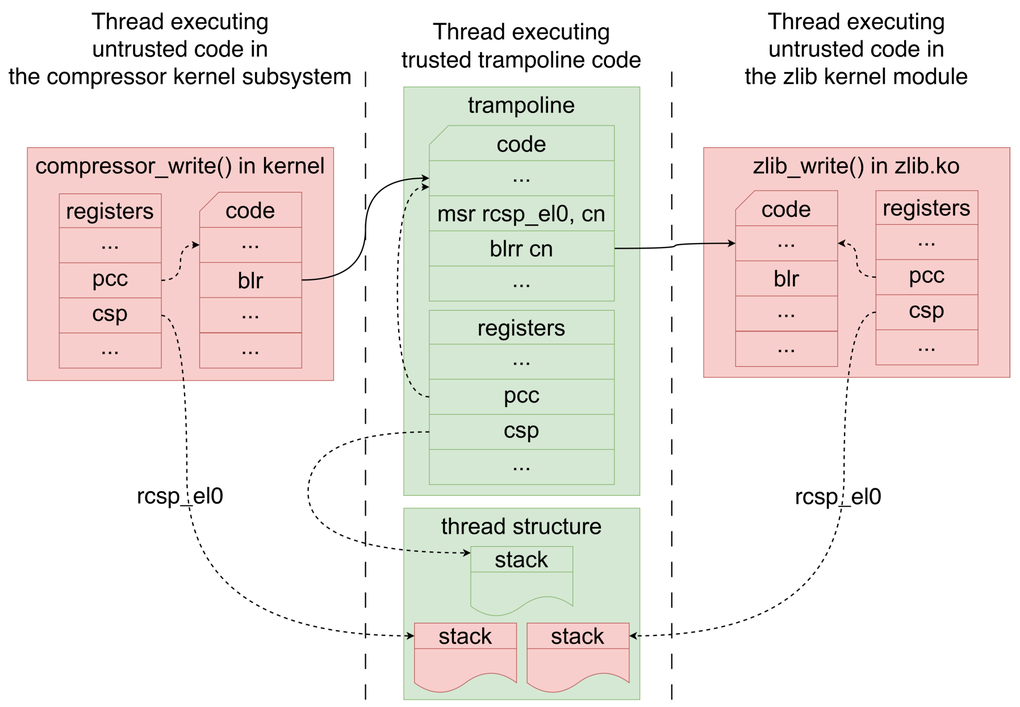
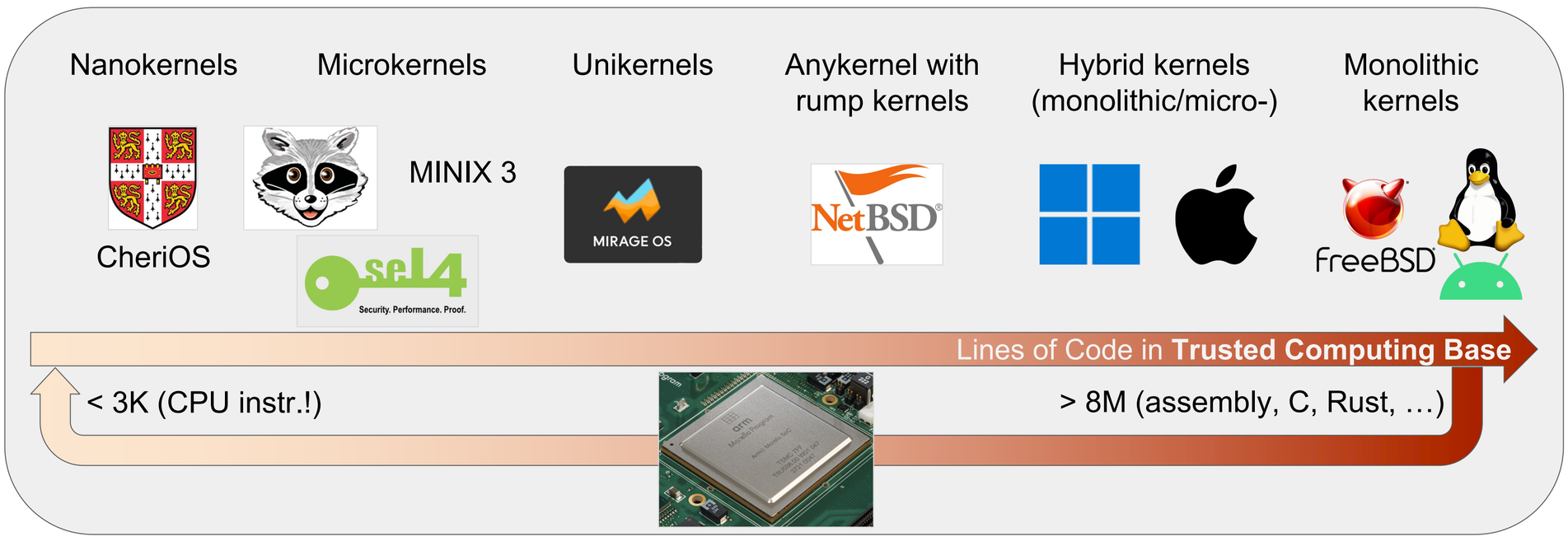


Figure 2. An example function call between compartments using a trampoline. The `blrr` instruction branches into the Restricted mode using a capability without the Executive permission bit set making the `csp` register refer to `rcsp_el0` instead of `csp_el1`



Hypothesis: Hardware-assisted compartmentalization with CHERI can significantly reduce the risk of vulnerabilities being exploited in a monolithic kernel with minor implementation and performance costs.

Contact Konrad.Witaszczyk@cl.cam.ac.uk

UNIVERSITY OF CAMBRIDGE

Approved for public release; distribution is unlimited. Sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-24-C-B047 (“DEC”) as part of the I2O CPM research program. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.