seL4 Microkernel and Container Viability

Cody Malick
malickc@oregonstate.edu

May 1, 2017

Abstract

This paper outlines the viability of implementing a container system on top of a mature microkernel, namely seL4. It briefly outlines the goal of this paper, general design goals of a microkernel architecture, seL4 history, and challenges microkernel architecture face. It also outlines what a container system is at a basic level, and what requirements from the kernel are needed for implementation. Lastly, it contains the conclusions on whether or not a microkernel architecture is suited for implementation and use of a container system.

Contents

Introduction

Over the past several years, the container architecture has garnered an intense following amongst both developers and systems engineers alike. With the widespread adoption of containers, comes the need for more thoughtful implementation regarding security and performance. Modern container systems have put great effort into securing their implementations through container isolation and minimizing resource use.[1] These settings are on by default to protect the organizations using contianer architecture. The primary concern is that it is the nature of a shared kernel that system integrity relies on all containers being secure. If the parent operating system is compromised, then every container and, more importantly, all other services running on containers are now compromised.

This problem problem stems intrinsically from the fact that operating security is hard. Linux, while some of the smartest minds in the world have worked incredibly hard to secure it, runs into new vulnerabilities regularly. This has to do with the fact that the Linux Kernel is extremely large. As of Linux 4.9, the kernel sports about eighteen million lines of code.[**?**] While a very large portion of this is comprised of device drivers maintained in the kernel itself, there are still millions of lines of code dedicated exclusively to the operating system functionality. While Linux is objectively a great operating system, it is difficult or nearly impossible to secure and prove that it can be secure.

This problem is one that is attempting to be solved by the community building and supporting the seL4 microkernel. Microkernels, originally conceived by Per Brinch Hansen and later pioneered by Johen Liedtke, are minimalist implementations of operating system kerenels.[2] The seL4 microkernel in particular aims to provide a small (roughly seven thousand lines of code), provide provable security for the operating system through formal verification, and provide the performance benefits postulated by Johen Liedtke.[3]

Microkernel Architecture

Minimal Design

Container Basics

Container Requirements

Feasibility

References

[1] Docker. (2016, August) Introduction to container security. [Online]. Available: https://www.docker.com/sites/default/files/WP_IntrotoContainerSecurity_08.19.2016.pdf

[2] H. H. M. H. J. L. S. S. J. Wolter. (1997, October) The performance of u-kernel-based systems. [Online]. Available: os.inf.tu-dresden.de/pubs/sosp97

[3] Gernot. (2016, September) Frequently asked questions. [Online]. Available: https://wiki.sel4.systems/FrequentlyAskedQuestions