Embracing Open Firmware in HPC for Faster and More Secure Provisioning

LA-UR-20-25698

Abstract

Firmware is the first piece of software that is run when a computer powers on and its primary job is to initialize the computer's hardware, determine which operating system (OS) to boot, and boot that OS. The firmware in most motherboards shipped from hardware vendors, both consumer and commercial, is often closed-source and contains many insufficiently-audited drivers. With the ubiquity of the Unified Extensible Firmware Interface (UEFI) specification in firmware implementation, firmware has become even more complex, often assuming the role of an OS. For instance, Intel's EDKII firmware, the GRUB bootloader, and the Linux kernel all have their own network, filesystem, and USB drivers. All of these redundancies complicate and lengthen the boot process, and present a greater attack surface which can bypass the checks of even the OS. Here, it is shown that by replacing proprietary vendor firmware with a Linux kernel, a project which has reputable drivers, undergoes heavy scrutiny, and is updated frequently, one can get a more secure, flexible, and resilient boot on HPC nodes. This allows for node provisioning to occur earlier in the boot process in firmware rather than after the OS boots, as well as greater control over how provisioning is implemented. Since Linux supports many platforms, this allows more homogeneous firmware images to be managed in a scalable way by the maintainer of the nodes instead of relying on the vendor. This project applies existing efforts to make firmware open source, including LinuxBoot and u-root, to HPC and large clusters. Being able to control what the firmware does provides one of the last missing pieces of open source in an HPC stack.