Trustworthy Board Management Software

Daniel Schwyn
ETH Zurich
Zurich, Switzerland

Ben Fiedler
ETH Zurich
Zurich, Switzerland

Roman Meier
ETH Zurich
Zurich, Switzerland

Michael Giardino[*]
ETH Zurich
Zurich, Switzerland

David Cock
ETH Zurich
Zurich, Switzerland

Timothy Roscoe
ETH Zurich
Zurich, Switzerland

Most modern computing platforms are so complex that they need a separate embedded system to manage them. These systems are referred to as (Base-)Board Management Controllers or BMCs. BMCs handle power and clock sequencing, and manage firmware for other components on the board. Furthermore, they usually offer remote management capabilities (e.g. console and firmware updates) over remote shells or web-interfaces. This collision (combination?) of absolute power, implicit trust and exposure to (the outside) makes BMCs simultaneously the root of trust and a significant threat vector.

Traditionally, BMC software is proprietary, closed-source firmware with no way to independently inspecting it for correctness, and which frequently exhibits security vulnerabilities. Open-source projects like OpenBMC or u-BMC are a step in the right direction, but still fall short of the high requirements for their security (e.g. CVE-2019-6260, CVE-2020-14156, CVE-2023-25507).

Moreover, addressing these system security concerns is meaningless if the BMC does not manage the system safely. This challenge starts with simply powering up the system: the power and clock distribution networks must be configured correctly and in the right order. Getting this power sequence wrong, or failing to react appropriately to runtime faults like over-temperature can destroy expensive hardware.

We experienced these challenges first-hand when engineering the firmware stack for Enzian[1], a heterogeneous platform for systems research. The lack of published literature on board management software design led us to conclude there is, as yet, no principled approach to BMC design.

To address these challenges, we developed a declarative model for power networks and can synthesize correct power sequences from these models [3]. Furthermore, we are investigating regulator driver synthesis for correctly executing such a generated sequence. This comprises work on model checking specifications of I^2C stacks but also generating C code, and even hardware implementations of I^2C controllers from these specifications [2]. The goal is to generate as much of the power management software as possible from models. We are confident that this approach generalizes to other BMC tasks like firmware management.

However, the safety properties of our service stack must be preserved even in the presence of untrusted BMC components, e.g. a remote management server, and so BMC services need to be securely isolated from each other, while still being able to interact safely. To obtain these isolation guarantees, we are building a high-assurance BMC stack on the solid basis of a verified separation kernel: seL4. Critical components like the power management stack will be implemented as native seL4 tasks while less trusted components are isolated in VMs. This cyber-retrofit approach also allows for reusing parts of existing BMC solutions like OpenBMC.

Building a high-assurance BMC stack with strong guarantees for critical components fills the dire need for safer and more secure platform firmware in modern computing platforms, but it also provides an excellent use-case for research on how to build trustworthy systems and push their complexity boundaries.

Validating our ideas on the Enzian platform provides us with insight into the following research questions: (1) What formal models of hardware allow us to generate correct software for real systems or verify its correctness? (2) How can communication channels be designed to preserve isolation guarantees between participants of different trust levels in real world computer systems? (3) How can correctness guarantees from hardware be extended to software, e.g. by generating an I^2C controller from a specification extracted from the hardware schematic?

[1] David Cock et al. "Enzian: An Open, General, CPU/FPGA Platform for Systems Software Research". URL: https://doi.org/10.1145/3503222.3507742.

[2] Lukas Humbel et al. "A Model-Checked I^2C Specification". URL: https://doi.org/10.1007/978-3-030-84629-9_10.

[3] Jasmin Schult et al. "Declarative Power Sequencing". URL: https://doi.org/10.1145/3477039.

[*] Now at Huawei Research

Eurosys'23, May 8–12, 2023, Rome, Italy
2023.