Abstract

The availability of new processors with more processing power for embedded systems has raised the development of applications that tackle problems of greater complexity. Currently, the embedded applications have more features, and as a consequence, more complexity. For this reason, there exists a growing interest in allowing the secure execution of multiple applications that share a single processor and memory. In this context, partitioned system architectures based on hypervisors have evolved as an adequate solution to build secure systems.
One of the main challenges in the construction of secure partitioned systems is the verification of the correct operation of the hypervisor, since, the hypervisor is the critical component on which rests the security of the partitioned system. Traditional approaches for Validation and Verification (V&V), such as testing, inspection and analysis, present limitations for the exhaustive validation and verification of the system operation, due to the fact that the input space to validate grows exponentially with respect to the number of inputs to validate. Given this limitations, verification techniques based in formal methods arise as an alternative to complement the traditional validation techniques.

This dissertation focuses on the application of formal methods to validate the correctness of the partitioned system, with a special focus on the XtratuM hypervisor. The proposed methodology is evaluated through its application to the hypervisor validation. To this end, we propose a formal model of the hypervisor based in Finite State Machines (FSM), this model enables the definition of the correctness properties that the hypervisor design must fulfill. In addition, this dissertation studies how to ensure the functional correctness of the hypervisor implementation by means of deductive code verification techniques.

Last, we study the vulnerabilities that result of the loss of confidentiality (CWE-200 [CWE08b]) of the information managed by the partitioned system. In this context, the vulnerabilities (infoleaks) are modeled, static code analysis techniques are applied to the detection of the vulnerabilities, and last the proposed techniques are validated by means of a practical case study on the Linux kernel that is a component of the partitioned system.