CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Poster

Towards the design of a DSL to enable the secure Runtime Monitoring and Verification of Safety-Critical CPS

Giann Nandi

CISTER-TR-190606

Towards the design of a DSL to enable the secure Runtime Monitoring and Verification of Safety-Critical CPS

Giann Nandi

CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail:

https://www.cister-labs.pt

Abstract

Towards the design of a DSL to enable the secure Runtime Monitoring and Verification of Safety-Critical CPS

Giann Spilere Nandi[1]

[1] CISTER – Research Centre in Real-time & Embedded Computing Systems, Instituto Superior de Engenharia do Porto, Rua Alfredo Allen 535, 4200-135 Porto, Portugal - giann@isep.ipp.pt

Author Keywords. Runtime Monitoring, Security, Cyber-Physical Systems, Domain Specific Languages.

1. Context and Motivation

Safety-critical systems commonly face unpredictable and hostile environments, with emergent behaviors and with a growing number of external, malicious attackers. These are risk factors that should be taken into account during these systems design phases, but that is not always possible due to the overall complexity of the interaction between the systems and its external operational environment. Cyber-Physical Systems (CPS) are notable examples of practical implementations of safety-critical systems. Being able to guarantee that safety-critical CPS do not fail upon operation can easily become a huge challenge, depending on how complex the system is. Among the most promising approaches to reduce the complexity of designing safety-critical CPS are Runtime Monitoring (RM) (Watterson and Heffernan 2017) and Runtime Verification (RV) (Bartocci Et al. 2018), where monitors are generated and orchestrated in a software architecture that can be coupled to the target system, observe it during its execution, and identify aspects that were not foreseen during design phase, or that could not be proved to be absent via static verification methods. Monitors can be used to verify the correct functioning of a system by analyzing direct (and/or indirect) aspects of it. This can be especially useful when considering a security-oriented point of view, where monitors can identify possible security attacks to a system when exposed to the events taking place or the patterns of data being processed.

2. Related Work

However, when making use of monitors, one has to ensure that the use of such technique does not *1)* influence negatively in the security aspects of the original system by, for example, leaking information that should be kept private (Krieg Et al. 2012); *2)* affect the functional and non-functional requirements of the system, for instance, tasks scheduling requirements (Khan Et al. 2016). As for the state-of-the-art in the integration of security and functional aspects of monitoring generation, the design of monitoring architectures and the implementation of security aspects are performed separately, where no formal verification of such integration is performed. This can lead to problems of design and implementation, possibly affecting aspects of security, safety, dependability, and performance of the monitored system.

3. Project Proposal

As an attempt to approach such problems, we intend to investigate novel approaches that enable the verification of CPS via the coupling of the system with runtime monitoring architectures in such a way that security and safety aspects are considered already in the design/implementation phase, and that the associated monitors do not interfere with the original specifications and behavior of the target monitored system. For that, we follow along the lines of designing a new Domain Specific Language (DSL) that allows to express what monitoring functional and non-functional properties should be verified during runtime, as well as security and privacy requirements for the communication between the components being

monitored and the monitors, generating in this manner the code that implements the underlying runtime verification architecture and the code for the monitors and their coupling with the architecture.

The envisioned DSL will also consider in the core of its design, the generation of verification conditions that will ensure the satisfaction of the functionality of the monitors, the respect of their interactions with the system in terms of security and privacy properties, and the consistency of the underlying runtime monitoring architecture. These verification conditions will be submitted to external automatic formal verification tools like model checkers, SMT solvers, and security focused formal verification tools tool for verifying security related properties) to guarantee their correctness-by-construction. This exempts the system designer/developer from the necessity of worrying about the security aspects of what needs to be monitored in the system, being the compiler of the proposed DSL the entity responsible for ensuring the generation of a secure monitoring architecture.

Regarding more concrete details about how we envision the design of the DSL and associated tool support, our proposal is to focus on using/extending existing formal coordination languages (e.g., the Reo language), to specify runtime monitoring architectures for distributed CPS with privacy and real-time requirements. To the best of our knowledge this has never been address in any prior research effort. Aspects that are expected to be explored include the extension of the current set of Reo's connectors to the case of distributed systems, and also to enrich them with privacy related properties and explicit timing constraints to cover the verification of fundamental aspects related to real-time embedded computing. Once this is achieved, the objective is to select/extend a suitable formal semantics that is able to express the behaviors one can specify using the envisioned DSL and generate the verification conditions already mentioned.

References

Bartocci, E., Y. Falcone, A. Francalanza, and G. Reger. "Introduction to Runtime Verification." In Lectures on Runtime Verification: Introductory and Advanced Topics, edited by E. Bartocci and Y. Falcone, 1–33. Cham: Springer International Publishing, 2018.

Watterson, C., and D. Heffernan. "Runtime verification and monitoring of embedded systems." IET software 1, no. 5 (2007): 172–179.

Krieg, A., J. Grinschgl, C. Steger, R. Weiss, H. Bock, and J. Haid. "System side-channel leakage emulation for HW/SW security coverification of MPSoCs." In 2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS), 139–144. April 2012.

Khan, M. T., D. Serpanos, and H. Shrobe. "A rigorous and efficient run-time security monitor for real-time critical embedded system applications." In 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), 100–105. December 2016.

Acknowledgement

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UID/CEC/04234); also by the Norte Portugal Regional Operational Programme (NORTE 2020) under the Portugal 2020 Partnership Agreement, through the European Regional Development Fund (ERDF) and also by national funds through the FCT, within project NORTE-01-0145-FEDER-028550 (REASSURE).