chist-era

SUCCESS

Security Certification and Quantitative Modeling of Human-Centric, Security and Privacy Properties

Grant Agreement:	N/A
Project Acronym:	SUCCESS
Project Name:	SecUre aCCESSibility for the internet of things
Instrument:	CHIST-ERA Call 2015
Thematic Priority:	SPTIoT
Start Date:	1 December 2016
Duration:	36 months
Document Type[1]:	T (Technical Report)
Document Distribution[2]:	CO (Confidential)
Document Code[3]:	SUCCESS-Inria-PR-001
Version:	v1.0
Editor (Partner):	F. Kammüller (Inria)
Contributors:	Stefano Schivo, Ioana-Domnina Cristescu, F. Kammüller
Workpackage(s):	WP3
Reviewer(s):	M. Bozga
Due Date:	Month 36
Submission Date:	November 2019
Number of Pages:	8

Funded by EPSRC Engineering and Physical Sciences Research Council; Agence Nationale de la Recherche ANR; NWO Netherlands Organisation for Scientific Research

[1] MD = management document; TR = technical report; D = deliverable; P = published paper; CD = communication/dissemination.

[2] PU = Public; PP = Restricted to other programme participants (including the Commission Services); RE = Restricted to a group specified by the consortium (including the Commission Services); CO = Confidential, only for members of the consortium (including the Commission Services).

[3] This code is constructed as described in the H2020 Project Handbook.

Security Certification and Quantitative Modeling of Human-Centric, Security and Privacy Properties

Ioana-Domnina Cristescu[1]

[1]INRIA

REVISION HISTORY

Date	**Version**	**Author**	**Modification**
17.1.2020	1.0	F. Kammüller	Set up the delivery documentation
27.1.2019	1.1	M. Bozga	Final approval

APPROVALS

Role	**Name**	**Partner**	**Date**
approver	M. Bozga	UGA	27.1.2020

Contents

1 Executive Summary

Security certification and quantitative modeling with special regard to modeling the human are the key concepts that characterize the SUCCESS process: a process of *Security by Design*.

Security certification can already be initiated at the early stage of formal system specification using the Isabelle Infrastructure framework. The later phases further support attack refinement using rare event model checking and lead to an executable BIP model adhering to the security properties.

Quantitative modeling is at the centre of the rare events modelchecking process used to enforce the high level specification onto a BIP component based model while refining it with respect to security using attack trees. In addition, timed automata are used for further quantitative modeling involving time and allowing modelchecking using UPPAAL.

The current document summarizes the results obtained within the Project refering to the earlier technical deliverable Milestone M2.2 that provides more detail on the process in a Conceptual Framework Draft.

2 Security Certification and Quantitative Modeling

Initially, we briefly summarize the Project outcomes regarding security certification before leading over to a summary of the results on Quantitative Modeling.

2.1 Security Certification of Human-Centric, Security and Privacy Properties

Security certification can already be initiated at the early stage of formal system specification using the Isabelle Infrastructure framework. Within this framework a formal infrastructure specification with actors and policies can be created, security and privacy goals can be proved but also attacks can be derived using the attack tree calculus [5]. Furthermore, very recently a feedback process for security refinement within the Isabelle Infrastructure framework has been defined, called the Risk-Refinement Loop [8, 7].

A distributed IoT architecture has been designed for SUCCESS using the Isabelle Insider framework and it has been illustrated how GDPR compliance can be proved for this system specification [6]. This architecture is intended to be used for the Pilot 2.

To enforce the security goals in a component based system implementation the formal system specification can then be represented in a dedicated IoT language (including the attacker). A translator allows to generate BIP models that can be executed in the subsequent phases of the SUCCESS process in the BIP simulation engine [2] leading to refined and security tested models. The generation of code from the verified BIP system thereby becomes possible.

A more detailed account of both phases of the SUCCESS process is given in Deliverable D2.1 and Milestone M6(M2.2).

Since recently the Inria team has lost its two leading members (they are on extended leave), it is questionable whether the originally planned code generation from BIP models to Javascript can still be implemented. This was a task internally assigned to Albert Cohen and potential collaborators funded from the project. Alternative routes to executable code are investigated including code generation from Isabelle to Scala.

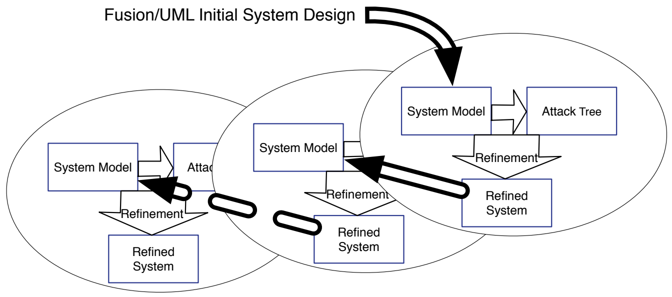

Figure 2.1: Refinement-Risk-Loop iterates design, risk analysis, and refinement

2.2 Quantitative Modeling for Rare Events Modelchecking

A BIP system, obtained from an IoT model, is analyzed using statistical model checking (SMC). The analysis we do is a quantitative one, that is, we ask what is the probability of a successful attack on an BIP system given an attack tree. Monte Carlo is a SMC method which consists of sampling the executions of an IoT system and computing the probability of a successful attack based on the number of executions for which the attack was successful. However, the evaluation may be difficult if a successful attack is rare, meaning that it occurs with a probability 10^{-3} or smaller. Monte Carlo requires a large number of simulations for a correct estimate of a rare event which is not always feasible. We therefore use *importance splitting* [4], an SMC method developed for rare events.

To implement this we rely on BIP, for which an execution engine is developed and maintained [1]. The IoT model is translated into a BIP model and the attack tree into a BIP monitor. The two form a BIP system. The execution engine of BIP produce executions which are the input of Plasma [3], our model checker.

This part is described in detail in Milestone 6 (M2.2).

2.3 Quantitative Modeling for Timed Automata

We refer to the project publication [9].

3 Conclusion

Bibliography

[1] A. Basu, S. Bensalem, M. Bozga, J. Combaz, M. Jaber, T.-H. Nguyen, and J. Sifakis. Rigorous component-based system design using the bip framework. *IEEE Software*, 28(3), 2011.

[2] D. Beaulaton, I. Cristescu, A. Legay, and J. Quilbeuf. A modeling language for security threats of iot systems. In F. Howar and J. Barnat, editors, *Formal Methods for Industrial Critical Systems*, pages 258–268, Cham, 2018. Springer International Publishing.

[3] B. Boyer, K. Corre, A. Legay, and S. Sedwards. Plasma-lab: A flexible, distributable statistical model checking library. In K. Joshi, M. Siegle, M. Stoelinga, and P. R. D'Argenio, editors, *Quantitative Evaluation of Systems*, pages 160–164, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[4] C. Jégourel, A. Legay, S. Sedwards, and L. Traonouez. Distributed verification of rare properties with lightweight importance splitting observers. *CoRR*, abs/1502.01838, 2015.

[5] F. Kammüller. Attack trees in isabelle. In *20th International Conference on Information and Communications Security*, volume 11149 of *LNCS*. Springer, 2018.

[6] F. Kammüller. Formal modeling and analysis of data protection for gdpr compliance of iot healthcare systems. In *IEEE Systems, Man, and Cybernetics, IEEE SMC'18*. IEEE, 2018.

[7] F. Kammüller. Attack trees in isabelle and the risk-refinement loop, 2019. Invited for a Special Issue to *Computers and Security* Elsevier. Extended Version of ICICS 2018 paper including aspects of SPTIoT paper.

[8] F. Kammüller. Combining secure system design with risk assessment for iot healthcare systems, 2019. Workshop of Security, Privacy, and Trust in the IoT, SPTIoT'19, colocated with IEEE PerCom'19.

[9] S. Schivo, B. M. Yildiz, E. Ruijters, C. Gerking, R. Kumar, S. Dziwok, A. Rensink, and M. Stoelinga. How to efficiently build a front-end tool for uppaal: A model-driven approach. In K. G. Larsen, O. Sokolsky, and J. Wang, editors, *Dependable Software Engineering. Theories, Tools, and Applications*, pages 319–336, Cham, 2017. Springer International Publishing.