From Secure Business Process Modeling to Design-Level Security Verification (Artifact Paper)

Qusai Ramadan[1], Mattia Salnitri[2], Daniel Strüber[1], Jan Jürjens[1] and Paolo Giorgini[2]

[1] University of Koblenz-Landau, Koblenz, Germany
{qramadan, strueber, juerjens}@uni-koblenz.de
[2] University of Trento, Trento, Italy
{mattia.salnitri, paolo.giorgini}@unitn.it

1 Introduction

In this paper, we present the artifact submission for our paper of the same name [1], to be presented at the MoDELS conference 2017 in Austin, TX. Our submission includes the model transformation from SecBPMN2 to UMLsec models as well as four example models from the Air Traffic Management System case study. We explain the process of using the transformation, and the verification of the generated UMLsec models using the CARiSMA tool.

In Fig.1, we show the processes and the input/output artifacts of our approach. Our approach consists of two processes, the first process provide an automated model transformation from SecBPMN2 models to corresponding UMLsec structural diagrams (i.e., deployment and class diagrams), using the model transformation language Henshin and its associated toolset. This process includes a set of transformation rules (.henshin) files and a Java code for rules orchestration. In the henshin files we define a set of Henshin transformation rules. The rules are defined graphically and applied to the input model (i.e., SecBPMN models) via an interpreter engine provided by Henshin. The output of this process is UMLsec models, and as a byproduct, our transformation rules create a trace model. The trace model links the SecBPMN2 and UMLsec models.

In the second process, we use CARiSMA to automatically verify the generated UML models against UMLsec policies. The output of this process is a text file the summaries the results of the verification process.

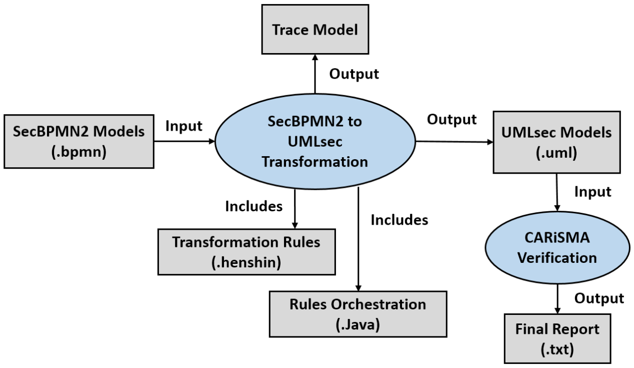

Fig. 1: The processes and the input/output artifacts.

2 Usage

Prerequisite. We recommend using Eclipse Neon, Modeling Tools distribution, with an installed nightly build of Henshin and CARiSMA. These softwares plug-ins can be installed on your Eclipse (Help **–>**Install New Software...) from the the follwing update sites: for CARiSMA use (http://carisma.umlsec.de/updatesite), while for Henshin one can use (http://download.eclipse.org/modeling/emft/henshin/updates/release).

Performing the transformation. To execute the transformation from SecBPMN2 to UMLsec models, please follow the following instruction. More details are available in the ReadMe file.

- Import our project package "myexample" to your local Eclipse workspace.
- Right click on the main class *BpmnToUml.java***–>** Run As *JUnit Plug-in Test* to perform the transformation. By default, our transformation takes the *example1.bpmn* file as input. To change the input file, first copy the name of one of the BPMN files that are provided in *myexample* **–>** *src* **–>** *my.example* directory. Second, find the following line of code (**public static final String EXAMPLE = "example1.bpmn";**) in the *BpmnToUml.java* file and replace the file name *"example1"* with the name of the selected BPMN file.
- After executing the *BpmnToUml.java*, you should see console output informing you about the generation process.
- The results of the transformation process (.uml file) will be stored to the *myexample* directory. The name of the UML file is *Transformed_serialized_profile*.

Performing the verification. In this step, we use CARiSMA checks to verify the generated UML models against UMLsec security policies.

- Right click in the *Project Explorer* view **–>** New **–>** Other **–>** CARiSMA **–>** Analysis –> Next -> in the dialog select the file that is generated from the last step (i.e., Transformed_serialized_profile.uml) and then click finish.
- From the dialog, click on *add checks to the list* icon –> select the check that you want to perform (e.g., secure links UMLsec check and secure dependency UMLsec check) then click run. For abac policy, you have to select both *RABACsec: Create transformation input* and *RABACsec: Use transformation input* checks. The former allows you to select the role that you want to verify his accessibility to the system operations, while the later return the set of operations that the selected role has an access to them. More details about the execution of CARiSMA checks are provided in the ReadMe file. Other information can also be found in the user manual of CARiSMA. After installing CARiSMA, the manual is available under:Help –> Help Contents –> CARiSMA.
- The result of the verification is provided in the *Analysis Results view*. One can also right click on the result and select *create a report for selected analysis*. The report will be stored to the *myexample* directory.

References

1. Q. Ramadan, M. Salnitri, D. Strüber, J. Jürjens, and P. Giorgini, “From Secure Business Process Modeling to Design-Level Security Verification,” (Accepted).