Design of a Core-based Specification Language for Dynamic RBAC Policies

Master Thesis Exposé

Fabian Kittler
Ilmenau University of Technology
fabian.kittler@tu-ilmenau.de

October 29, 2014

During the last decades, IT systems have greatly changed. Many new methods and paradigms were introduced. One of these paradigms were security policies. Security policies are a set of rules, observance of which shall provide certain security properties. Policies are usually composed in natural speech. Because of this informal notation, they are unsuitable for formal verification of security properties. Consequently, the policies have to be converted to a formal model. Thereafter, these security models can be analyzed for certain properties. There are already diverse tools for this purpose. In order for the tool to understand and process the security models, the model has to be formally specified. For this step, commonly specification languages are used. They supply the tools with a machine-readable input of the models.

WorSE [AKP14] is an example for these tools. WorSE is a workbench for engineering and analyzing security models. There are many different models such as HRU or dynamic ABAC and RBAC models [HRU76, SCFY96]. Nowadays, formal security models are often base on a state machine. In this context, the idea of core-based security models [Pöl14] was introduced. In order to reduce the error rate, WorSE assists a security engineer at the creation of security models. Additionally, it provides the functionality to analyze these models. It is also possible to import and export security models in a XML schema.

1 Problem Statement

WorSE and the XML schema are currently not suited for the specification of core-bases security models. Moreover, the specification of models in the XML schema is at an improper level of abstraction for security engineers. They rather work with different models components and mathematical constructs. This results in a gap between the language paradigms. In different areas of computer science, such kind of gaps is found as well. There, domain specific specification languages are used to fill the gap. The same approach can be applied here likewise.

2 Goal

To tackle this problem is the goal of this thesis. A better approach is to provide a specification language which can be used by security engineers, so that the engineers can specify the model in their domain and do not have to care about the format the tools need as input.

The aim of this thesis is to design a specification language for core-based security models. The language shall be integrated into WorSE. Since WorSE can only import models given in an XML schema, the output of the compiler must be an XML schema.

The thesis will only comprise the specification of dynamic RBAC models because, as mentioned before, WorSE cannot understand core-base models at the present time. Dynamic RBAC models are frequently used and offer through role-hierarchy (RBAC1) and restrictions (RBAC2) a high expressiveness. However, it is to be designed in a way to allow the future integration of other security models. The dynamic RBAC model is selected because it represents realistic use case scenarios and is therefore interesting to security engineers. Nevertheless, the language shall generally be able to specify core-based models since the analysis of core-based model is to be integrated into WorSE.

3 Approach

At first, the model components and paradigms of dynamic RBAC models and the core-based models will be analyzed to collect the requirements for the specification language. In the next step, different specification languages shall be looked at to identify useful constructs and used paradigms with respect to the collected requirements. On this basis, a specification language for dynamic RBAC models will be designed. This requires the specialization of the state machine and the extension vector of the core-based model. The specific components and the operations for the transition function have to be defined. Certainly, such a specification language must offer all needed model components.

In order to build a compiler, not only the specification language has to be defined but also the input format that is expected by WorSE has to be analyzed. For the language definition the Extended Backus–Naur-Form (EBNF) could be used. This enables the automatic parsing of the language. Afterwards the compiler has to be implemented according to the definitions.

In the end, the resulting specification language and the compiler have to be evaluated. The evaluation of the paradigms and the language itself have to be distinguished. The usability can be tested through the specification of sufficient large models. In this process, the requirements can be verified and it can be explained how they were reached. The focus of this thesis is the specification of dynamic RBAC models but the specification language shall support core-base models. It will be evaluated what steps are necessary to integrate a different security model and how complex it is. To evaluate the compiler, it shall be integrated into WorSE and analyzed for runtime and complexity.

References

[AKP14] Peter Amthor, Winfried E. Kühnhauser, and Anja Pölck. WorSE: A Workbench for Model-based Security Engineering. *Computers & Security*, 42(0):40–55, 2014.

[HRU76] Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. Protection in Operating Systems. *Communications of the ACM*, 19(8):461–471, August 1976.

[Pöl14] Anja Pölck. *Small TCBs of Policy-controlled Operating Systems*. PhD thesis, Ilmenau University of Technology, Ilmenau, Germany, May 2014.

[SCFY96] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, February 1996.