PANIC – Plugging Data Leak Detection into the Camunda Modeler

Jasmin Türker[1] and Thomas S. Heinze[2]

[1] Institute of Computer Science
Friedrich Schiller University Jena
jasmin.tuerker@uni-jena.de
[2] Institute of Data Science
German Aerospace Center (DLR)
thomas.heinze@dlr.de

Abstract. We present PANIC, a certified data leak detection tool, implemented as a plugin for the Camunda Modeler. With the help of our tool, process designers can detect data leaks instantaneously and interactively. The current prototype supports arbitrary BPMN processes with data logic implemented as inline scripts of a Groovy subset.

1 Introduction

Business processes are often subject to regulations governing the confidentiality and privacy of the information that is dealt with. To avoid accidental disclosure of sensitive or private data, automated process auditing can be a useful method. In this paper, we present a static analysis plugin for the popular *Camunda Modeler*[3] tool for modeling business processes in BPMN. Our plugin implements a data flow analysis, which allows for detecting data leaks at process design time.

A data flow analysis analyzes the propagation of data in a business process. Our analysis thereby distinguishes between data objects of two different sensitivity levels: *low*, representing objects originating from an insensitive data source, and *high*, representing objects from a sensitive data source. In order to guarantee the absence of data leakage, the analysis has to check that no *high*-level object is propagated to an untrusted, that is *low*-level sink. In contrast, any flow of objects between equal levels or from *low*-level source to *high*-level sink is allowed.

Our plugin is thus based on a two-level security model for mandatory access control. Technically, the plugin therefore implements a unified points-to/taint analysis [3] to track the flow of sensitivity-labeled data objects inside a business process. The analysis has been formally proven correct using a mechanized proof, such that the analysis is also an instance of a certified data flow analysis. Based on our previous work detailing the analysis and its verification [4], the contribution of this paper is the prototype implementation of the PANIC analysis plugin.

[3] https://camunda.com/

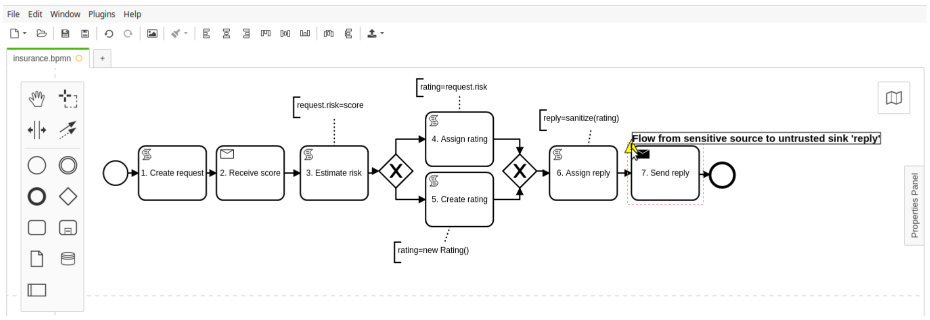

Fig. 1. Screenshot of the PANIC data leak detection plugin

2 Data Leak Detection Plugin

In the Camunda Modeler, process data of BPMN business processes can be modeled by process variables, messages, and (inline) scripts. While a number of programming languages are supported for defining inline scripts in the Camunda platform, our plugin currently only supports a small subset of the Groovy language, where Groovy is a JVM-based language providing an object-oriented data model. A process activity can be decorated as script task and attached a Groovy script, which gets access to a local copy of the process variables and messages, and to an API for committing local changes back to the variables and messages. To distinguish between *low*- and *high*-level objects, we defined a BPMN extension to decorate an activity as source of sensitive data or as untrusted sink. This applies in particular to inbound and outbound message activities, which can be marked as sources and sinks by the process designer, respectively.

Task of the data leak detection plugin is to reason about the flow of labeled data objects as defined by process variables, messages, and inline scripts. The plugin is therefore implemented as a client-server application, where the client is the plugin itself, providing the Camunda Modeler GUI extensions and bindings, and the server implements the actual analysis in the OCaml programming language. Note that the analysis implementation is directly generated from its correctness proof and only once at plugin installation time. The analysis can run in just milliseconds each time a BPMN process is modified in the Camunda Modeler, thus supporting interactive and instantaneous modeling and analysis.

3 Demo

The implementation of the PANIC plugin is available online[4]. Using the plugin, a process designer can easily identify flaws in her process model with respect to potential data leaks. Figure 1 shows a simple BPMN process for a fictive risk assessment of a health insurance company, where a sensitive medical record is

[4] `https://gitlab.com/fu63rov/camunda-static-analysis-plugin`

propagated to an unstrusted sink. As can be seen, the plugin identifies this flaw in the process model and notifies the process designer. In the demo, we will show how a process designer can interact with the plugin, while creating a BPMN process model step-by-step, to identify and repair potential data leaks.

4 Related Work

Mandatory access control is a standard security model for data confidentiality [2], where access is granted based on policies using security classification levels such that access from a lower classified object to a higher classified object is prohibited. Process modeling and analysis with respect to mandatory access control has been a topic in the business process domain in recent years. Analysis usually applies the Petri net formalism, e.g., mapping data leak detection to the reachability problem as in [1]. However, we are not aware of any work which considers actual process data in terms of variables, messages, and even scripts modifying them, nor do we know tool support in process modeling applications like Camunda.

5 Conclusion and Future Work

In this demo paper, we have presented a prototype implementation for a Camunda Modeler plugin which allows a process designer to detect potential data leaks while creating BPMN process models. In its current version, the plugin is able to analyze data logic implemented as inline scripts of a subset of the Groovy language. In future work, we would like to extend the plugin to cover full Groovy as well as other programming languages supported by the Camunda platform. In particular, we wish to integrate support for the Java language and respective delegate classes, which are usually implementing a process' service tasks. Besides, we are working on making the analysis more precise, increasing precision by means of flow-sensitivity.

References

1. Accorsi, R., Lehmann, A., Lohmann, N.: Information leak detection in business process models: Theory, application, and tool support. Inf. Sys. **47**, 244–257 (2015)
2. Denning, D.E.: A Lattice Model of Secure Information Flow. Comm. ACM **19**(5), 236–243 (1976)
3. Grech, N., Smaragdakis, Y.: P/Taint: Unified Points-to and Taint Analysis. PACMPL **1**(OOPSLA), 102:1–102:28 (2017)
4. Heinze, T.S., Türker, J.: Certified Information Flow Analysis of Service Implementations. In: SOCA 2018. pp. 191–198. IEEE (2018)