Information Systems 47 (2015) 244–257

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Information leak detection in business process models: Theory, application, and tool support ☆

Rafael Accorsi [a,*], Andreas Lehmann [b], Niels Lohmann [b]

[a] University of Freiburg, Department of Telematics, 79098 Freiburg, Germany
[b] Universität Rostock, Institut für Informatik, 18051 Rostock, Germany

ARTICLE INFO

Available online 13 January 2014

Keywords:
Business process security
Software and process engineering
Automated analysis

ABSTRACT

Despite the correct deployment of access control mechanisms, information leaks can persist and threaten the reliability of business process execution. This paper presents an automated and effective approach for the verification of information flow control for business process models. Building on the concept of place-based non-interference and declassification, the core contribution of this paper is the application of Petri net reachability to detect places in which information leaks occur. Such a feature allows for the use of state-of-the-art tool support to model-check business process models and detect leaks. We show that the approach is sound and complete, and present the Anica tool to identify leaks. An extensive evaluation comprising over 550 industrial process models is carried out and shows that information flow analysis of process models can be done in milliseconds. This motivates a tight integration of business process modeling and non-interference checking.

1. Introduction

Business processes (BPs) specify how activities are executed to provide a service; for instance steps in a supply-chain or profile updates in customer relationship management. In doing so, they handle sensitive data that must remain confidential. However, design flaws in BPs design may lead to a violation of such confidentiality requirements [2]; that is, they can cause leaks.

Leaks happen when data or information flows from a secret domain to a public domain. The former is called *data leak*; the latter is referred to as *information leak*. A data leak is a direct but illegal access to a data object. An information leak concerns the fact that unauthorized subjects can infer secret information. Although certification standards (e.g., ISO/IEC 27001 [3] and TCSEC [4]) demand the identification of both kinds of leaks, current state of the art mainly provides mechanisms to detect data leaks (e.g., [5–10]).

In this paper, we focus on the use of formal methods to identify information leaks in BP models. BPs are the main asset of companies as they describe their value chain and fundamentally define "the way businesses are done". In this scenario, information leaks can have catastrophic consequences for the business owner, be it for legal or financial reasons. *Information flow control* provides a powerful abstraction to reason about information leaks [11]. Assuming that the BP model under analysis is split into two logical security domains (*high* for secret, *low* for public), a BP model is assumed "secure" with regard to information leaks if it enforces *non-interference*; that is, the actions in the high domain do not produce an observable effect (*interference*) in the low domain. Details on the security domains follow in Section 2. By showing non-interference for a BP model one thus ensures that subjects in the low domain cannot deduce information about the high domain, thereby guaranteeing its confidentiality and isolation.

We consider the Petri net representation of BP models as a basis for the analysis. For this, mappings from common

☆ Extended and thoroughly revised version of [1].

* Corresponding author.
E-mail addresses: rafael.accorsi@iig.uni-freiburg.de (R. Accorsi), andreas.lehmann@uni-rostock.de (A. Lehmann), niels.lohmann@uni-rostock.de (N. Lohmann).

0306-4379/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2013.12.006

modeling languages, such as WS-BPEL, BPMN and EPC, exist [12]. Subsequently, the activities — denoted as Petri net transitions — are separated into high and low domains. The analysis of these models is carried out with *place-based non-interference* (PBNI) [13]. PBNI is an approach to encode and reason about *structural non-interference* (and hence information flow control) in Petri nets. The rationale is that specific types of places in the net encode different non-interference properties; that is, they denote information leaks. In showing the absence of such places in a Petri net, one rules out structural interferences.

The current approach [14] and tool support [15,16] for information flow analysis of BP models exhibit the following drawbacks: *Firstly*, the decision procedures to detect these places are based on the generation and analysis of the *complete* state space of the model, which due to the state space explosion renders a very inefficient approach for complex BP models. *Secondly*, formal proofs of the decision procedure are missing — in particular soundness and completeness properties. The lack of these guarantees weakens the security guarantees provided by tool-support, because results may contain false-negatives, for instance that a BP model is considered secure even though information leaks exist.

Contributions: This paper reports on the following contributions:

- We introduce an approach for the information flow analysis of BP models based on the reachability problem for Petri nets [17]. By reducing the analysis to reachability, we obtain a decision procedure based on standard Petri net reasoning, hence inheriting a broad spectrum of verification tools and techniques.
- We prove that the approach based on reachability is sound and complete.
- We present the design and implementation of *Anica*, a novel tool for information flow analysis of BP models. Anica employs Petri net analysis and state space reduction methods to generate and analyze only relevant parts of the state space. Its realization employs the model-checking tool LoLA [18].
- We evaluate Anica on a BP repository comprising over 550 industrial models. The information flow analysis of each BP takes only fractions of a second. Anica analyzes the entire repository, whereas other tools relying on complete state space exploration may fail for complex BPs.

This paper substantiates that the detection of information leaks in BP models can be carried out in an effective, push-button manner. Process modelers can employ these techniques to enforce non-leak security guarantees "by design" before process deployment. We show that the well-founded information flow analysis of industrial BPs can in fact be done on-the-fly, as we demonstrate in [19]. This opens the possibility of security analysis *during* BP editing, or at run-time upon the event of ad hoc process changes. In doing so, we contribute to the reliably secure modeling of flexible BP models.

Practical relevance: In the security jargon, the aforementioned information leaks are a consequence of *covert channels* [20] induced by the faulty structure of BP models. The detection of covert channels (and, hence, potential leaks) is required in various certification standards. For example, the TCSEC's "Part B: Mandatory Protection" demands the "analysis of covert channels and their bandwidth "[4]. Similarly, ISO/IEC 27001 Section 12.5 "Security in development and support processes" prescribes that "checks should be made for information leakage for example via covert channels" [3]. Concrete audit and certification procedures for SAS 70 and SAS 117 equally demand security guarantees [21, Chapter 24]. Several papers expose the risks of covert channels in fields related to BP management and deployment, e.g., Cloud Computing environments [22], virtualization [23] and Web Services [24]. The threat of leaks is considered imminent [25] and real [26, Chapter 8], thereby motivating our work and attesting its practical relevance.

In contrast to information leak detection, the *enhancement* of processes (e.g., with process rewriting [27]) to deter information leaks is usually not required [3]: the correction of one detected covert channel can in turn create new covert channels. The introduction of explicit declassification transitions appears to be an effective way of repairing business processes [19].

Approach overview: The proposed approach and tool support *Anica* take (the Petri net representation of a) BP model produced by a process modeler as input and produces a certificate containing the set of places where information leaks may occur. This process comprises two main steps: *Firstly*, the activities in the model are labeled with the corresponding security domain (namely: "high" for secret and "low" for public). This step can be carried out manually, or different strategies may be employed to automatically suggest labels to activities, depending on the goal and scope of analysis. The running example used to illustrate the approach employs, e.g., a strategy for the analysis of concurrent process executions (see [28] for strategies and their definitions). *Secondly*, Anica takes the labeled Petri net and automatically creates a series of reachability problems, whereas the underlying model-checking tool *LoLA* solves these problems and returns the corresponding witnesses. Fig. 8 illustrates the approach.

Organization: For completeness, Section 2 revisits the concept of PBNI and a small example. Section 3 presents the approach to detect non-interference based on reachability, and shows its main properties. Section 5 introduces Anica and Section 6 presents its evaluation. Section 7 compares our contribution with related work and Section 8 summarizes the lessons learnt and indicates further work.

2. Information flow control in Petri nets

This section provides the formal basis necessary to reduce the detection of interferences in BPs to a reachability problem. We firstly introduce the multi-level security model [29] used to reason about interferences. Secondly, we define the necessary Petri net background, then revisit *place-based non-interference* (PBNI) [13] to capture leaks as interferences.

Multi-level security model: We employ a multi-level security (MLS) model to capture and detect leaks as interferences