A Trustworthy Usage Control Enforcement Framework

Ricardo Neisse, Information Systems Quality (ISQ), Fraunhofer IESE, Kaiserslautern, Germany
Alexander Pretschner, Department of Computer Science, TU München, Garching, Germany
Valentina Di Giacomo, Engineering Ingegneria Informatica, Rome, Italy

ABSTRACT

Usage control policies specify restrictions on the handling of data after access has been granted. The authors present the design and implementation of a framework for enforcing usage control requirements and demonstrate its genericity by instantiating it to two different levels of abstraction, those of the operating system and an enterprise service bus. This framework consists of a policy language, an automatic conversion of policies into enforcement mechanisms, and technology implemented on the grounds of trusted computing technology that makes it possible to detect tampering with the infrastructure. The authors show how this framework can, among other things, be used to enforce separation-of-duty policies. The authors provide a performance analysis.

Keywords: Framework, Policy Enforcement, Security, Trusted Computing, Usage Control

INTRODUCTION

Usage control requirements stipulate the handling of data after access is granted and, as such, generalize access control requirements. Typical requirements include “notify when message leaves system”, “don’t delete for five years”, “delete data within thirty days”, and also actions one and two on a specific item must not be performed by the same individual”. The handling of data is reflected by system-level events at different levels of abstraction including the operating system (events: system calls), enterprise service buses (ESB; events: messages), windowing systems (X11; events: network packets), and so on. From this perspective, usage control policies hence stipulate allowed and disallowed sequences of events. These policies can be both detective and preventive in nature. Detective policies stipulate what is allowed and what is not allowed. The respective enforcement strategy is optimistic: if the policy is violated, not too much harm will happen, and the violation can be punished, compensated, or sometimes even undone. Preventive policies, in contrast, prevent violations from happening.

In this paper, which is an extended version of one of our previous publications (Neisse, Pretschner, & Di Giacomo, 2011), we present a usage control policy enforcement framework.

DOI: 10.4018/jmcmc.2013070103
that was developed as part of an EU-funded research project MASTER (MASTER, 2012). In the context of this project, our framework was used to demonstrate two application scenarios in a hospital and an insurance company. Requirements in these scenarios include Chinese Wall, Separation of Duty (SoD), and privacy protection through identity anonymization. Chinese Wall policies are required in an insurance company to prevent consultants working for competing companies from having access to sensitive information. SoD and privacy protection are required in the hospital scenario to anonymize patient data and to minimize the risk of fraud by preventing doctors from prescribing and dispensing medicine without involving a nurse or another doctor. Because of space restrictions, we concentrate on a SoD example in this paper.

The more general setting is as follows. Business processes are formalized as sequences of events. Control objectives (“more than one person should be involved in a drug prescription and dispensation to prevent fraud”) are defined in specification-level policies. Specification-level policies define control objectives, but they do not specify how the objectives can be achieved. This can, in general, be done in different ways: by blocking (e.g., forbid dispensation if the user that prescribed is the same trying to dispense medicine) or by executing additional actions (send a notification to an auditor requesting confirmation that the dispensation is not a fraud). The concrete choice is stipulated in so-called implementation-level policies. The usage control enforcement framework that we describe in this paper has a focus on implementation-level policies.

Our framework can be used for specification and enforcement of implementation-level policies at different levels of abstraction. To illustrate this flexibility we show an implementation that instantiates our framework at the operating system and ESB levels. In this implementation, prescription and dispensation are services running in an ESB. The motivation is that SoD policies enforced at the ESB level can be circumvented by a malicious user by directly modifying the files that contain the prescription and dispensation records. Moreover, in principle, it is simple to circumvent policy enforcement components from our framework, e.g., by killing the respective processes. As a consequence, we need to make sure that this cannot happen, or at least detect if it has happened. Since root users – the extreme case of potentially malicious insiders – usually can do everything they desire, we include in our framework components that use trusted computing technology to create tamper-proof integrity measurement logs. In the case of an audit, root users will then have to justify why specific changes to the system have been applied. For space reasons we do not provide here the security and performance evaluation of the integrity measurement components and refer to a previously published paper that describes this technology (Neisse, Holling, & Pretschner, 2011):

- **Research Problem:** The problem we tackle is the design and implementation of a usage control framework that makes it possible to specify and enforce detective and preventive implementation-level usage control policies, that can be instantiated to different levels of abstraction, and that provides protection against tampering;

- **Contribution:** We present such a system. In contrast to comparable approaches, our framework is based on a formal semantics, is more expressive because it supports temporal and cardinality operators, has been implemented and applied to real-world example scenarios, and provides protection against tampering with trusted computing technology;

- **Organization:** This paper is organized as follows. We first introduce the language we adopt to specify usage control policies. After this introduction, we describe the policy enforcement components and architecture. We then introduce the guarantees we provide using trusted computing technology followed by the presentation of two prototype implementations to en-
Related Content

Quality of Experience Models for Multimedia Streaming
www.igi-global.com/article/quality-experience-models-multimedia-streaming/47328?camid=4v1a

Distributed Approach for QoS Guarantee to Wireless Multimedia
www.igi-global.com/chapter/distributed-approach-qos-guarantee-wireless/17076?camid=4v1a

A Distributed and Scalable Solution for Applying Semantic Techniques to Big Data
www.igi-global.com/article/a-distributed-and-scalable-solution-for-applying-semantic-techniques-to-big-data/129000?camid=4v1a
Category-Based Interfaces for Mobile Search
[www.igi-global.com/chapter/category-based-interfaces-for-mobile-search/133750?camid=4v1a](www.igi-global.com/chapter/category-based-interfaces-for-mobile-search/133750?camid=4v1a)