COMPUTERS & SECURITY 55 (2015) 193–234

Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

Security solution frames and security patterns for authorization in distributed, collaborative systems

Anton V. Uzunov [a,*], Eduardo B. Fernandez [b], Katrina Falkner [a]

[a] School of Computer Science, The University of Adelaide, Adelaide, South Australia 5005, Australia
[b] Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL 33431, USA

ARTICLE INFO

Article history:
Available online 28 August 2015

Keywords:
Security engineering
Software engineering
Security patterns
Authorization
Access control
Distributed systems
Distributed systems security
Security solution frames

ABSTRACT

The design of an authorization infrastructure is one of the most important aspects of engineering a secure software system. Unlike other system types, distributed systems – and especially distributed collaborative systems – can require custom, fine-grained authorization models and enforcement approaches that are able to take into account a range of semantic subtleties. In this paper we present a comprehensive, pattern-oriented software engineering approach to authorization for general distributed systems – with particular applicability to distributed collaborative systems – that allows developers to build custom, application-specific conceptual authorization models in a simple yet extensible manner, and to make informed decisions regarding their enforcement in software, as well as how their supporting rule/policy infrastructure should be designed. Our authorization approach is embodied in two *instances* of a new pattern-based security engineering construct called a *security solution frame*, which groups together related patterns – both security "product" and micro-process patterns – in different sub-structures, horizontally and vertically, for a single high-level security policy (in our case authorization and policy management). By applying specific micro-process patterns in each solution frame, developers are guided in using relevant "product" patterns to progressively construct a distributed authorization infrastructure – from abstract concepts toward concrete designs, via a number of levels of abstraction implying solution refinement and corresponding to stages of the development life-cycle. The summary-form "product" patterns encapsulated in each frame also help developers to form a holistic, "global" view when analyzing existing infrastructures. We illustrate and evaluate the proposal in the context of greenfield system development by applying our solution frames to design the authorization infrastructure of a (new) distributed system for secure file sharing and collaborative editing; and also use our solution frames to briefly analyze and capture the design decisions underlying two existing distributed authorization infrastructures: one based on UCON for collaborative Grid systems and another based on ZBAC for SOA-based systems.

* Corresponding author. Tel.: +61 873897252.
E-mail addresses: anton.uzunov@outlook.com (A.V. Uzunov), ed@cse.fau.edu (E.B. Fernandez), katrina.falkner@adelaide.edu.au (K. Falkner).
http://dx.doi.org/10.1016/j.cose.2015.08.003
0167-4048/

1. Introduction

Security is becoming an increasingly important aspect of distributed systems (Anderson, 2008; Belapurkar et al., 2009; Pourzandi et al., 2005) and of collaborative systems in particular (Lu et al., 2009; Zhang et al., 2008; Zou et al., 2008), where multiple groups of users and components interact not only across physical but often across organizational boundaries as well. One of the most important aspects of engineering a secure distributed, collaborative system is to determine which actions by system entities are and are not allowed – i.e. the *design* of an authorization infrastructure (Lodderstedt et al., 2002; Ren et al., 2005). Related to this and of equal importance is the architectural *analysis* of existing authorization infrastructures, which forms a basis for the latter design activity and also often follows it in the form of documenting design decisions.

Traditionally, models for authorization have been classified into discretionary (DAC), mandatory (MAC), role-based (RBAC) (Bertino and Sandhu, 2005), and, more recently, attribute-based (ABAC) (De Capitani di Vimercati et al., 2008; Sandhu, 2012). A number of arguments exist in the literature, however, which suggest that traditional authorization models are not always appropriate or sufficient for distributed settings (Blaze et al., 1999; Kane and Browne, 2006; Park and Sandhu, 2004), and certainly not for distributed collaborative settings with fine-grained authorization requirements (Edwards, 1996; Lu et al., 2009; Sikkel, 1997; Sikkel and Stiemerling, 1998; Tolone et al., 2005; Zhou, 2008).

In their survey of collaborative authorization models, Tolone et al. (2005) stipulate a number of characteristics which such models should possess, the most relevant of which can be summarized as follows:

- Authorization models should be generic and should permit rights "to be configured to meet the needs of a wide variety of cooperative tasks and enterprise models" (Tolone et al., 2005). It should be possible to specify rights based on various information, such as roles of users, contexts etc.
- Authorization models should allow the sharing of objects (environments, information, resources etc.) of any type and at different levels of granularity.
- Authorization models should permit, as required, for the specification and modification of authorization rules at runtime "depending on the environment or collaboration dynamics" (Tolone et al., 2005).

Naturally, a number of models that satisfy some or even all of these requirements can be found not only in the references already cited, but also in a wide variety of publications containing models for general, distributed and collaborative settings, whose number is so great, in fact, that one could fill a whole compendium if one were to enumerate just a fraction of them. Such models are continually being created to satisfy the new authorization requirements posed by the development of new systems (Barker, 2009).

While these models are valuable additions to the growing body of security knowledge, they also give rise to two important (even if speculative) questions. The first of these is whether the existing authorization models can take into account the semantics of the collaborative system to which they are applied. Since different systems have different needs, it is unlikely that a single model will be universally suitable, unless it is itself customizable. Indeed, the requirements stipulated above collectively imply that distributed, collaborative applications require custom authorization models (cf. Shen and Dewan, 1992) that can take into account a particular system's semantics. The second question pertains to whether the existing authorization models, many of which are theoretical, are actually being used in practice. More importantly, the question can be reformulated as whether software engineering teams, which often take a rather scornful attitude to incorporating security during development (see Hein and Saiedian, 2009; McGraw, 2002; Whyte and Harrison, 2011), are sufficiently interested in taking up and realizing the often formal descriptions of the models in everyday collaborative software, and sufficiently experienced to do so. After all, the vast majority of publications on authorization originate from academia, not the industry, and hence emphasize formal, theoretical and purely scientific values.

While the first question could be addressed via increasingly advanced authorization models and meta-models, the second question is not so easy to answer. In fact, addressing the first question by way of more complex models would only exacerbate the situation leading to the second question. With respect to both questions, one would be pressed to find a comprehensive, software engineering approach to authorization that guides security-inexperienced software developers (the predominant workforce – Whyte and Harrison, 2011) from the abstract concepts of authorization to concrete solutions that can be used when *designing* authorization infrastructures for general distributed and especially distributed collaborative systems. There is a similar dearth of software engineering approaches and detailed frameworks for comprehensively *analyzing* the design decisions underlying existing distributed authorization infrastructures.

In this paper, we seek to address the latter points and fill what we believe is an important gap in the literature, by providing a comprehensive software engineering approach (process and framework) for designing and analyzing distributed authorization infrastructures. Our approach is embodied in two *instances* of a new pattern-based security engineering construct called a *security solution frame*, which groups together security "product" patterns (Fernandez, 2013; Schumacher et al., 2006) and security micro-process patterns (Uzunov et al., 2015a) in different sub-structures, horizontally and vertically, for a single high-level security policy. Security solution frames[1] are a generic, methodology-agnostic security engineering artifact with broad applicability, and represent our first original contribution. Using the patterns from the first of our authorization frames, one can build custom, application-specific conceptual models suitable for a range of general distributed and distributed collaborative systems satisfying the authorization requirements set out by Tolone and colleagues, in a simple, efficient and extensible manner as befits pattern-based approaches. This is how we address the first problem outlined at the outset of the introduction, namely, lack of approaches for building up custom conceptual authorization

[1] Security solution frames should not be confused with Jackson problem frames, for which see, e.g. Schmidt (2010).