Computer Standards & Interfaces 36 (2014) 748–758

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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

Enterprise security pattern: A model-driven architecture instance

Santiago Moral-García [a,b,*], Santiago Moral-Rubio [c], Eduardo B. Fernández [d], Eduardo Fernández-Medina [e]

[a] Produban, Santander Bank, Boston, MA, USA
[b] Kybele Research Group, Dept. of Computer Languages and Systems II, Rey Juan Carlos University, Madrid, Spain
[c] BBVA Group, Madrid, Spain
[d] Secure Systems Research Group, Dept. of Comp. and Elect. Eng. and Comp. Science, Florida Atlantic University, Boca Raton, FL, USA
[e] GSyA Research Group, Dept. of Information Technologies and Systems, University of Castilla-La Mancha, Ciudad Real, Spain

ARTICLE INFO

Available online 30 December 2013

Keywords:
Secure cloud computing
Model driven architecture
Enterprise security architecture
Security pattern
Enterprise security pattern

ABSTRACT

To secure their information assets, organizations should seek support from enterprise security architectures. Security patterns are a good way to build and test new security mechanisms, but they have some limitations related to their usability. In previous work, we defined a new type of security pattern called *Enterprise Security Pattern*. The main objective of these patterns is to provide an instance of model-driven architecture, which offers a solution to recurring problems that have to do with information systems security. In recent years, the hiring of Software as a Service (SaaS) from cloud providers has become very popular. There seem to be many advantages of using these services, but organizations need to be aware of a variety of threats, as well as being prepared to handle them. In another work undertaken previously, we defined an enterprise security pattern called *Secure Software as a Service* (*Secure SaaS*), which the organizations could apply to protect their information assets when using SaaS. In this paper, we present different instances of the solution models of the enterprise security pattern *Secure SaaS*, aiming to verify the risks that an organization would assume if each of the instances were deployed. With this approach, we intend to show how the design decisions adopted when performing the transformations between the solution models can have a direct impact on the security provided by the pattern.

1. Introduction

Technological advances are currently improving many aspects related to the development and design of information systems, thus entailing an increase in the complexity of enterprise security architectures. This in turn brings about a rise in the number of attacks [15]. The biggest problem that we have encountered in the design of enterprise security architectures is that different information security engineers provide a variety of solutions to the same problem. Noting the large number of security engineers who may work for the same organization, it is seen to be necessary to find a carefully thought and proven set of security guidelines to assist engineers when designing enterprise security architectures.

An approach which is considered promising for this purpose is the use of security patterns [20]. Security patterns provide guidelines to support the construction and evaluation of new security mechanisms [5,17]. The use of security patterns helps to incorporate security principles when building secure systems [16]. However, they have some limitations:

- They are small units of defense. They can only handle one (or a few) threat. Considering the number of threats that can affect current information systems, a security designer should select an extensive set of security patterns when building secure systems.
- It is necessary to have different versions of the same pattern, one for each architectural level. As the building of secure systems needs an extensive set of security patterns, this fact increases the complexity when a security designer is trying to select a pattern.
- Several instantiations of a pattern may have common aspects, but the designer has to recognize them; failing to do so may cause unnecessary redundancies.

In previous work [13], and bearing these limitations in mind, we defined a new type of security pattern called *Enterprise Security Pattern*. The main objective of these patterns is to provide a top-down strategy based on models for defining enterprise security architectures across different levels of abstraction, including their technological implementation. Organizations would be able to use enterprise security patterns to select a global security strategy for a specific business process. This provides their designers with a set of security guidelines thereby standardizing the design and building of the enterprise security architecture for that process. In addition, security engineers could, on one hand, manage security elements included in the different abstraction models separately. On the other hand, they would be enabled to perform automatic transformations between them. This fact would make it easier for the designer to select and tailor the enterprise security patterns when s/he is building enterprise security architectures.

* Corresponding author at: 2 Morrisey Boulevard, DORCHESTER, 02125 MA, USA.
E-mail addresses: smoralga@produban.us, santiago.moral@urjc.es (S. Moral-García), santiago.moral@bbva.com (S. Moral-Rubio), ed@cse.fau.edu (E.B. Fernández), Eduardo.FdezMedina@uclm.es (E. Fernández-Medina).

0920-5489/$ – see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.csi.2013.12.009

The practice of outsourcing business functions has been around for decades. In recent years, it has become very popular as an online software service [4]. Online software delivery is now conceived and defined as Software as a Service (SaaS). SaaS focuses on separating the *possession* and *ownership* of software from its *use* [19]. The advantages of using SaaS seem to be numerous, because this online service model may prove cheaper than owning and maintaining an in-house IT system [11]. Companies expect to save money on support and upgrade costs, IT infrastructure, IT personnel, and implementation. However, this new environment brings some threats that organizations need to deal with. In another previous piece of work [14], we defined an enterprise security pattern called *Secure Software as a Service* (*Secure SaaS*), the aim of which was to provide proven and reliable enterprise security architecture that assures the information assets when organizations decide to outsource their applications.

In this paper, which has evolved from the work mentioned previously [14], we present different instances of the solution models of the enterprise security pattern *Secure SaaS*, aiming to find out the risks which an organization would incur if each of the instances were deployed. With this approach, we intend to show how the design decisions adopted when performing the transformations between the solution models can have a direct impact on the security provided by the pattern.

The remainder of this paper is organized as follows. Section 2 provides a description of the concept of enterprise security patterns, including their context and solution model. Section 3 presents the enterprise security pattern *Secure Software as a Service* (*SaaS*). Section 4 shows different transformations for each of the solution models. Finally, Section 5 presents some conclusions and future work.

2. Enterprise security patterns

The biggest problem that we have encountered when designing enterprise security architectures is that different information security engineers provide different solutions to the same problem, because each one has his or her own guidelines, or follows his or her own criteria. In addition, the number of iterations when developing the architecture and the reusability level between one design and the next one depends on the experience and quality of the engineer, rather than on the method used.

One of the basic principles of information security is that there are no formal mechanisms to verify the robustness of enterprise security architectures. This means that the number of valid architectures is low and can only be validated by taking account of the time that they have been operating in the industry without being violated. Similarly, organizations should use only enterprise security architectures that have been proved on the basis of their history of absence of incidents. Given these problems, it becomes necessary to find a careful thought and proven set of security guidelines for designing enterprise security architectures; it needs to be a set that is based on the historical validation of the mechanisms used. This set of security guidelines would help to reduce significantly the risk of designing architectures which are apparently useful, but which actually have security weaknesses.

Enterprise security patterns offer solutions to recurring problems related to information systems security, promoting the reusability of designs when developing enterprise security architectures. They provide a top-down strategy that is based on models for defining enterprise security architectures across different levels of abstraction, including their technological implementation. An enterprise security pattern combines a wide range of items. These describe generic enterprise security architectures that provide some security properties for a set of information assets in a specific context. To do this, enterprise security patterns put together in one cohesive pattern all the elements included in the enterprise security architectures: (i) the *information assets* to be protected, (ii) the *context* in which these assets are found, (iii) the *threats* associated with the assets, (iv) the *security policies*, *patterns*, *mechanisms* and *technologies* used to stop these threats, and (v) the *stakeholders* and *systems* involved in the solution. Enterprise security patterns are not intended to replace security patterns. They use the latter, integrating them into a more comprehensive pattern that can handle a greater number of threats.

By using enterprise security patterns, the Chief Information Security Officer of an organization could select a global security strategy for a specific business process, providing its designers with an optimal and proven security guideline. This would in turn standardize the design and building of the enterprise security architecture for that process. By using enterprise security patterns, security engineers could also, on one hand, manage security elements included in the different abstraction models separately. As well as this, they would be able to perform automatic transformations between them. This fact would enable the designer to carry out the selection and tailoring of security policies, patterns, mechanisms and technologies with greater ease when building enterprise security architecture for a given business process. To aid in the understanding of these patterns, we will go on to give a description of their *context model* and *solution model* in the following paragraphs.

2.1. Context model

The context of enterprise security patterns describes the generic environment under which these patterns should be applied. This context may include (i) the type of information assets to protect (*data*, *applications*, and *code and configuration*), (ii) the security realms where the assets are stored, (iii) the security policies associated with the information assets, and (iv) the general features of who (*customers*, *employees*, or *technical users*) or what (*systems*) will access them. We define here the *security realms model* and the *security policies model*.

2.1.1. Security realms model

Other work [1,18] uses the concept security domain to refer to this term. However, the term "security domain" has been adapted to have different meanings in different areas, such as physical security, and JBoss. We have thus decided to call it *security realm*, in order not to confuse the reader.

Security realms can be defined as logical and discrete entities that partition the enterprise network. The main purpose of these realms is to standardize enterprise security and thus reduce the cost, user delay, and administrative overheads of redundant security procedures. The main characteristic of security realms is that each of them has in common the same security policies. An instance of a security realm results in one or more enterprise sub-networks.

When classifying the security realms, we have taken into account the *Types of Realms* (TR) that can be found in an enterprise network, as well as who manages each of those realms, i.e., a *Trust Level* (TL). The classification of Security Realms (SR) that we propose here can be defined as *SR*: $TR \times TL$. The specific realms can be adjusted to fit different types of applications; what matters here is that we use a classification of this type. Table 1 shows with an "✓" each of the eighteen security realms provided in our classification.

Table 1
Classification of security realms.

		Trust levels		
		Managed	Externally managed	Public
Types of realms	Customer	✓	✓	✓
	Employee	✓	✓	✓
	Technical user	✓	✓	✓
	Development	✓	✓	–
	Data	✓	✓	–
	Bastion	✓	✓	–
	Transport	✓	✓	✓