Web Services Security Configuration in a Service-Oriented Architecture

Takeshi Imamura Michiaki Tatsubori Yuichi Nakamura
IBM Research, Tokyo Research Laboratory
1623-14, Shimotsuruma, Yamato,
Kanagawa 242-8502, Japan
+81-46-215-{4479, 4958, 4576}
{imamu, tazbori, nakamury}@jp.ibm.com

Christopher Giblin
IBM Research, Zurich Research Laboratory
CH-8803 Rüschlikon, Switzerland
+41-1-724-8478
cgi@zurich.ibm.com

ABSTRACT

Security is one of the major concerns when developing mission-critical business applications, and this concern motivated the Web Services Security specifications. However, the existing tools to configure the security properties of Web Services give a technology-oriented view; only assisting in choosing data to encrypt and the encryption algorithms to use. A user must manually bridge the gap between the security requirements and the configuration, which could cause extra configuration costs and lead to potential misconfiguration hazards. To ease this situation, we came up with refining security requirements from business to technology, leveraging the concepts of Service-Oriented Architecture (SOA) and Model-Driven Architecture (MDA). Security requirements are gradually transformed to more detailed ones or countermeasures by bridging the gap between them by using best practice patterns.

Categories and Subject Descriptors

D.2.1 [Software Engineering]: Requirements/Specifications – methodologies, tools.

General Terms: Design, Security.

Keywords: Web Services Security, Service-Oriented Architecture, Model-Driven Architecture, security configuration, best practice pattern.

1. INTRODUCTION

Security is one of the major concerns when developing mission-critical business applications, and this concern motivated the Web Services Security specifications [1]. These specifications are very flexible so as to cover various security requirements. However, such flexibility contributes to usability issues. Users have to specify many detailed parameters, such as cryptographic algorithms and encryption keys.

In [2], we discussed a security configuration tool based on best practice patterns. The assumption there was that users should begin by considering security requirements at a higher abstraction level and then move onto technical details. In this paper, we further discuss the refinement of security requirements from business to technology, leveraging the concepts of Service-Oriented Architecture (SOA) [3] and Model-Driven Architecture (MDA) [4]. We analyze what security requirements can exist at each level and how the security requirements at different levels can be associated.

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

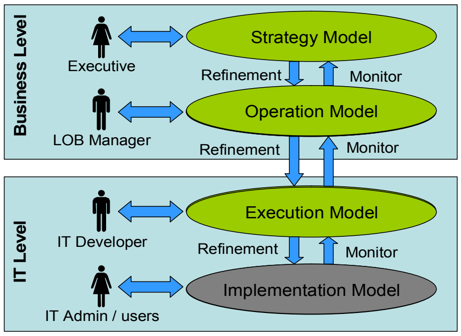

Figure 1. Service-Oriented Architecture

2. USING SOA FOR SECURITY

Figure 1 shows an SOA framework proposed by IBM. Business and IT levels are defined, and each level has two models. Each model is defined explicitly considering who is concerned with that model. These models are characterized and could be used to describe security requirements as follows:

- *Strategy model* – description of an organization's strategic goals, business design, and business objectives. → High-level rules: legislation, business practices, and corporate-level rules and guidelines.
- *Operation model* – compute-independent model of activities representing business processes and rules. → Constraints and rules with which business processes must comply. Some constraints can be validated at this level, and others have to be verified at a lower level.
- *Execution model* – platform-independent description of documents, flows, and connections to people, applications, and data sources, and their relationships. → Application-specific rules. We describe what security requirements must be met at this level.
- *Implementation model* – platform-specific model of IT infrastructure – hardware, system software, network infrastructure, and middleware. → Platform-dependent configuration. We describe how to achieve the above security requirements. In addition, the security infrastructure is taken into account.

3. BRIDGING BETWEEN SOA MODELS

Because there is no standard or even a proposed means for the strategy model, we start by considering the security description at the operation model. In other words, our initial goal is to come up with languages for representing security-related information at the operation, execution, and implementation models. Also, according to the MDA concept, we consider how to elaborate higher-level descriptions to lower levels.

3.1 Patterns in the Operation Model

The framework we propose is shown in Figure 2. An overview of the framework is as follows: a user in the top-level model is given a vocabulary list containing the terms that he usually uses. He is asked to describe his security requirements using this vocabulary. Then, the security requirements are gradually transformed to more detailed ones, for example, through communication with the users in each model, or by considering other requirements or restrictions from the environment. Once a certain degree of detailed security requirements have been obtained, countermeasures can be considered. Then, as was done for the security requirements, the countermeasures are gradually transformed into more detailed ones. Once a certain degree of detailed countermeasures have been obtained, they are described in some policy language. And finally, a user in the bottom-level model configures his system according to the description. Or the system might be directly configured without such a description.

There could be various ways to gradually transform security requirements or countermeasures. Here we use patterns. A user is asked to choose some of the patterns, which were derived from actual use cases, and apply them to his security requirements or countermeasures. Each pattern specifies a transformation rule from an upper level description to a lower one. According to the rules specified by the applied patterns, the security requirements or countermeasures are transformed. A user is asked to repeat this process until a certain degree of detailed security requirements or countermeasures have been obtained.

3.2 Patterns in the Operation Model

In [5], we focused on the security of Web Services messaging and classified pairs of security requirements and countermeasures in the execution model as patterns. To be concrete, we assumed that there are four types of security requirements; *message confidentiality*, *message integrity*, *non-repudiation of messages*, and *user authentication*, and enumerated countermeasures for each type of security requirement. We follow this classification here and try to classify patterns that derive security requirements in the execution model from those in the operation model.

Considering some examples in the real world, it seems possible to express them as terms used in access control technology, such as *readable*, *writable*, and *executable*. Patterns classified from this point of view are cataloged as follows:

1. *Synopsis* – provide readable message contents.
   Context – message content readability is collapsed.
   Solution – provide message contents that are not confidential from one (and may be confidential from others).

2-a. *Synopsis* – provide rewritable messages.
   Context – message rewritability by more than one is collapsed.
   Solution – provide messages with integrity.

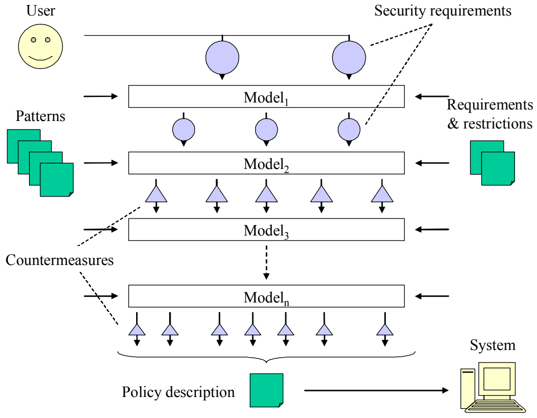

Figure 2. Framework for bridging between models

2-b. *Synopsis* – provide rewritable messages.
   Context – message rewritability by only one is collapsed.
   Solution – provide non-repudiated messages.

3. *Synopsis* – provide signed messages.
   Context – messages are invalidated.
   Solution – provide non-repudiated messages.

4. *Synopsis* – make operations executable.
   Context – operation executability is collapsed.
   Solution – provide authenticated message sources.

4. CONCLUSION

In this paper, we introduced SOA and MDA, and discussed their applicability to security configuration. Then we proposed a framework to refine security requirements from business to technology, leveraging the concepts of SOA and MDA. Future work could include the following:

- Extend the tool we developed in [2] so that it can support the operation model.
- Extend the framework we proposed so that it can start with risks or security objectives rather than security requirements.
- Develop a more user-friendly way to choose appropriate patterns from those available.
- Study the nature of overlaps and conflicts among patterns and develop a reasonable way to detect and address them.
- Study and embody potential directions to extend the framework, especially support of bottom-up information propagation and application to compliance technology.

5. REFERENCES

[1] OASIS. Web Services Security: SOAP Message Security 1.0, *OASIS Standard*, 2004.

[2] M. Tatsubori, T. Imamura, and Y. Nakamura. Best Practice Patterns and Tool Support for Configuring Secure Web Services Messaging, *IEEE International Conference on Web Services (ICWS)*, 2004.

[3] IBM. Service Oriented Architecture (SOA). *http://www.ibm.com/software/info/openenvironment/soa*

[4] OMG. Model Driven Architecture (MDA). *http://www.omg.org/mda*

[5] T. Imamura and M. Tatsubori. Patterns for Securing Web Services Messaging, *OOPSLA Workshop on Web Services and Service Oriented Architecture Best Practice and Patterns*, 2003.