A Method for Eliciting Security Requirements from the Business Process Models

Naved Ahmed and Raimundas Matulevičius

Institute of Computer Science, University of Tartu
J. Liivi 2, 50409 Tartu, Estonia
{naved, rma}@ut.ee

Abstract. In recent years, the business process modelling is matured towards expressing enterprise's organisational behaviour (i.e., business values and stakeholder interests). This shows potential to perform early security analysis to capture enterprise security needs. Traditionally, security in business processes is addressed either by representing security concepts graphically or by enforcing these security constraints. However, these approaches miss the elicitation of security needs and their translation to security requirements for system-to-be. This paper proposes a method to elicit security objectives from business process models and translate them to security requirements. The method enables early security analysis and allows developers not only to understand how to protect secure business assets, but it also contributes to alignment of the business processes with the technology that supports the execution of the business processes.

Keywords: Security in Business Processes, Business Process Modelling, Requirements Engineering

1 Introduction

Although the importance of introducing security engineering practices early in the development cycle has been acknowledged, it has been overssighted in business processes and targets the improvement of business function. The reason behind is that the business analysts are expert in their domain but having no clue about the security domain [10]. There has been several attempts to engage the relatively matured security requirements engineering in business processes. The majority of studies either focusses on the graphical representation of security aspects in business process models [8,10] or enforces the security mechanisms [7] or both [11]. These studies have neglected the security requirements elicitation. They analyse major problems when addressing security engineering in business process modelling. Firstly, security requirements are specified in terms of security architectural design (i.e., security control) and missing the rationale about the trade-offs of the security decision. Secondly, the requirement elicitation is either missing or haphazard: this leads to miss some critical security requirements. And finally, due to the dynamic and complicated nature of business processes

the studies only addresses varying aspects (i.e., authorization, access control, separation of duty or binding of duty) but not the overall security of business processes. These problems can be overcome by eliciting security objectives from the organizational business processes and by transforming them to the security requirements of the operational business processes where the technology supports the business processes execution.

Here we analyse the research question, *how to elicit security objectives from the business processes and to translate them to security requirements*? We propose a method consisting of two major stages. Firstly, it describes how to identify business assets and to determine their security objectives. Secondly, it supports eliciting security requirements from the operational business processes.

The rest of the paper is structured as follows. In Section 2 we introduce the illustrative example. Section 3 presents our method for eliciting security requirements from the business processes. Section 4 concludes the paper and presents some future work.

2 Land Management System

To perform the security requirements elicitation one needs to collect the knowledge of enterprise *value system*, including the *value chain* and the *business functions*. Fig. 1 illustrates a value chain for the LMS example. It organises the enterprise business functions and relates them to each other (as enterprise cooperates to achieve the business goals). In Fig. 2 we present a detailed workflow of Prepare Plan *p*rocess. The *p*rocess has two business partners (Lodging Party and Planning Portal) expressed as *s*wimlanes, while Registry is identified as an information system.

Fig. 1: Land Management Systems - Value Chain

Similarly to Prepare Plan, other sub-processes (e.g., Lodge Plan, Examine Plan, Approve Plan and Update Plan) are also expanded to the operational and conversation models. But in Section 3, we will present our proposal using the Prepare Plan process (as illustrated in Fig. 1 and 2).

3 Security Requirement Elicitation Method

In [2], we have presented a set of security risk-oriented patterns for securing business processes. Based on these patterns, in this section, we introduce a method (Fig. 3) to elicit security requirements as constraints that have to be respected when executing a business process. The first stage is dedicated to *business asset identification* and *security objective determination*. In the second stage, the *elicitation of security requirements* is done from the system's contextual areas.

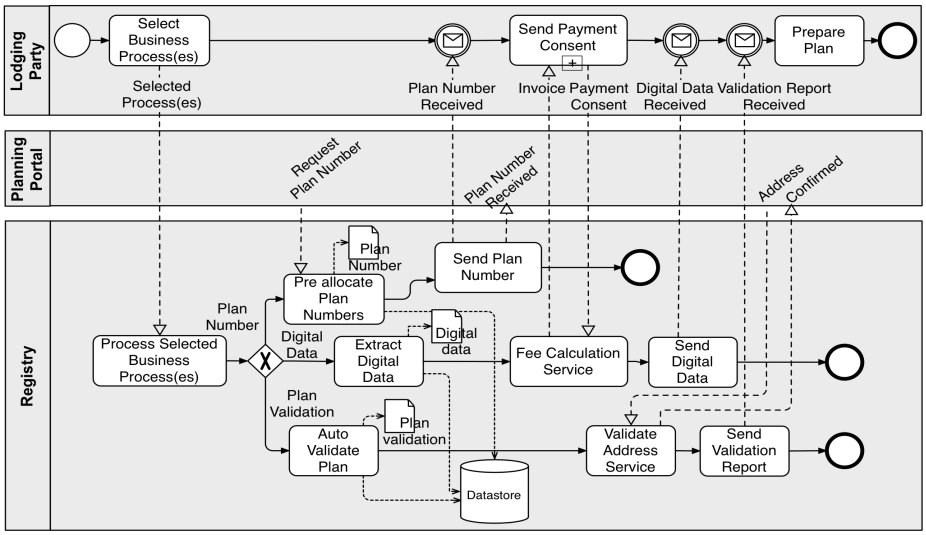

Fig. 2: Operational Business Process - Prepare Plan

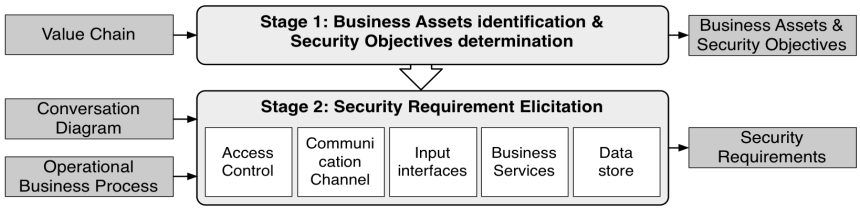

Fig. 3: Security Requirements Elicitation Method

3.1 Stage 1: Business Assets identification & Security Objectives determination

The first stage starts with the analysis of the *value chain*, which (*i*) gives an understanding of organisational processes and, thus, (*ii*) helps determine the assets that must be protected against security risks. In the LMS case the protected asset is Plan since it is the central artefact used in all the business activities (see Fig. 1). In terms of the security objective: *i*) Plan should be confidential, i.e., no unauthorised individual should read it and its relevant data; *ii*) Plan should be integral, i.e., the Plan and its relevant data should not be tempered; and *iii*) Plan and its relevant data should be available to the business partners at anytime.

3.2 Stage 2: Security Requirement Elicitation

At the second stage, the security requirements elicitation is performed at five contextual areas: *access control*, *communication channel*, *input interfaces*, *business services*, and *data store*. It is important to note that each artefact –*data*

or *process*– separately considered and protected at each contextual area, contributes to the security of business asset (i.e., Plan) identified at the first stage.

Access Control specifies how the business assets could be manipulated by individuals, applications or their groups. The major concern is to protect the confidentiality of identified business asset, in our example the Plan, when it is being manipulated by the IS asset, (i.e., the Registry). The security threat arises if the access to the Plan and its properties, like (Plan Number, Digital Data, and Plan Validation) is allowed to users without checking their access permissions. The risk event would: *i*) negate confidentiality of Plan, *ii*) lead to the Plan unintended use, and *iii*) harm the Registry's reliability.

A way to mitigate the security risk is the introduction of access control mechanism, for example the Role-Based Access Control (RBAC) model. A *r*ole (e.g., Lodging Party and Planning Portal modelled using ≪role≫ stereotype) is a job function within the context of organisation. *P*ermissions characterise role privileges to perform operations on the protected object. An *o*bject is a protected resource (i.e., Plan). An *o*peration is an executable set of actions that can change the state of the protected resource. For instance, Pre allocate Plan Numbers, Send Digital Data, etc are operations which manipulate properties Plan Number, Digital Data and Plan Validation (Fig. 4). Permissions specify the security actions –namely, *Create*, *Read* and *Update*– that the role can perform over the state of the protected resource. The following security requirements are defined:

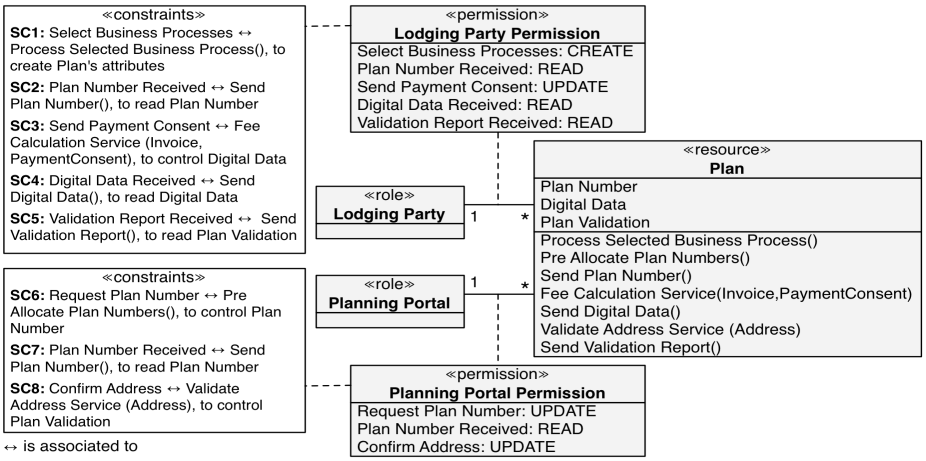

Fig. 4: RBAC Security Model - Prepare Plan Business Process

RQ1. Lodging Party should be able to:
1. create or initialize the Plan Number, Digital Data and Plan Validation.
2. read the Plan Number, Digital Data and Plan Validation.
3. update the Digital Data.

RQ2. Planning Portal should be able to:
1. update the Plan Number and Plan Validation.
2. read the Plan Number.

The security model (i.e., Fig. 4) defines how authorised parties should access the protected resources. However, it does not support capturing the concerns related to the separation of duties, binding of duties, and usage control [1]. The security requirements RQ3, RQ4 and RQ5 should be taken into consideration:

RQ3. Secured operations (e.g., Fee Calculation Service) should be performed by different users assigned to the same role.
RQ4. A sequence of secured operations (e.g., Pre allocate Plan Numbers and Send Plan Number) should be performed by the same user assigned to the role (e.g., Planning Portal).
RQ5. The system (i.e., Registry) should place constraints on how confidential data should be used by the roles (i.e., Lodging Party and Planning Portal).

RQ3 defines that there should exist at least two users in the Registry with the same role, to finish executing the task Fee Calculation Service: the first user issues the Invoice and the second user approves the Payment Consent. Requirement RQ4 highlights the concept binding of duties. Requirement RQ5 defines the security constraints for usage control; e.g., the Registry could potentially define constraints for Digital Data and Validation Report saying, that they remain valid for seven days. Elicitation of requirements RQ3-5 much depends on the concrete problem. They can't be captured from the business model and require involvement of business and/or security analysts.

Communication Channel is used to exchange data between business partners (e.g., Lodging Party and Planning Portal) and system (e.g., Registry). Here, data, like Selected Business Process(es), Payment Consent and etc, need to be protected when they are transmitted over the (untrusted) communication channel, i.e., Internet. The communication channel could be intercepted by the threat agent and the captured data could be misused (i.e., read and kept for the later use or modified and passed over) by the threat agent. This could lead to the loss of the channel reliability, and could negate the confidentiality and integrity of the Plan. To mitigate the risk, the requirements should be implemented for the Lodging Party and Registry and correspondingly for the Planning Portal and Registry:

RQ6. The server (e.g., Registry) should have the unique identity in the form of key pairs (public key, private key) certified by a certification authority.
RQ7. The client (e.g., Lodging Party and Planning Portal) should encrypt and sign the data (e.g., Selected Process(es), Plan Number, and other) using keys before sending it to the server (e.g., Registry).

A security requirements implementation could be fulfilled by the standard transport layer security (a.k.a., TLS) protocol [3] as illustrated in Fig. 5. As the first contact, the Lodging Party sends Registry a handshake message, which includes a random number. Following RQ6, the Registry responds with its public key and the information about the certification authority. After verification of the Registry's public key, the Lodging Party generates the secret and sends it to the Registry encrypted with the Registry's public key. The Registry then decrypt the secret using the private key and generates symmetric session keys. The keys enable Lodging party and Registry to establish a secure session for data exchange.

Following RQ7, encryption keeps the transmitted data (e.g., Selected Business Process(es), Payment Consent and etc) confidential and signing it ensures that the received data is not tempered. The secure communication continues until it is not explicitly terminated by Lodging Party or Registry.

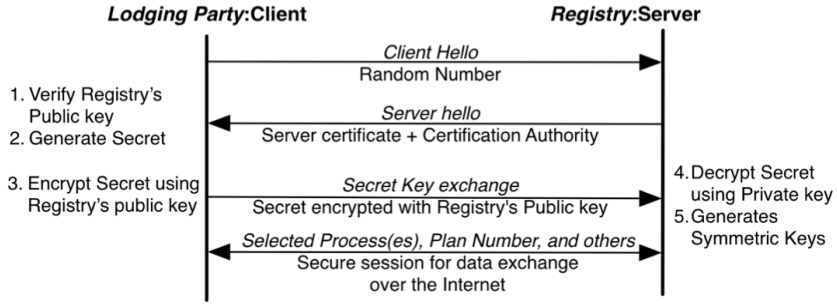

Fig. 5: TLS Protocol implementation, adapted from [3]

Input interfaces are used to input data submitted by business partners. In Fig. 2, we identify Process Selected Business Process(es) and Fee Calculation Service as input interfaces of Registry that receives the Selected Process(es) and Payment Consent from Lodging Party. The threat agent can exploit the vulnerability of the input interfaces by submitting the data with a malicious scripts. If happening so the availability and integrity of any activity (e.g., Send Digital Data) after the input interface (e.g., Fee Calculation Service) may be negated. To avoid this risk the following security requirements must be implemented for the input interface:

RQ8. The input interface (e.g., Fee Calculation Service) should filter the input data (e.g., Payment Consent).
RQ9. The input interface (e.g., Fee Calculation Service) should sanitize the input data (e.g., Payment Consent) to transform it to the required format.
RQ10. The input interface (e.g., Fee Calculation Service) should canonicalize the input data (e.g., Payment Consent) to verify against its canonical representation.

Input filtration [5] (RQ8) validates the input data against the secure and correct syntax. The string input should potentially be checked for length and character set validity (e.g., allowed and blacklisted characters). The numerical input should be validated against their upper and lower value boundaries. *Input sanitization* (RQ9) should check for common encoding methods used (e.g., HTML entity encoding, URL encoding, etc). The *input canonicalization* [5] (RQ10) verifies the input against its canonical representation.

Business Service is a task or activity executed within an enterprise on behalf of the business partner [6]. The goal is to guarantee availability of the business services. The business services, like Fee Calculation Service offered to Lodging Party, are provided by the server (e.g., Registry) through the communication channel. The threat agent may exploit the hosts in the channel and hack them because of the protocol (e.g., TCP, ICMP or DNS [4]) vulnerability; i.e., the ability to handle an unlimited number of requests for service. When receiving

simultaneously multiple requests, the server i.e., Registry, will not be able to handle them, thus, the services become unavailable. The successful denial of service attacks could also provoke the loss of partner's (e.g., Lodging Party and Planning Portal) confidence on Registry. To mitigate this risk, one could define three types of firewalls (see Fig. 6) – Packet Filter Firewall, Proxy Based Firewall and Stateful Firewall [12], and introduce the following requirements:

RQ11. Server (e.g., Registry) should establish a rule base (i.e., a collection of enterprise' constraints used by different firewalls) to communicate with the business partners (e.g., Planning Portal).
RQ12. Packet Filter Firewall should filter the business party's (e.g., Planning Portal's) address to determine if it is not a host used by the threat agent.
RQ13. Proxy Based Firewall should communicate to the proxy which represents the business service (e.g., Pre allocate Plan Number) to determine the validity of the request received from the business party (e.g., Planning Portal).
RQ14. State Firewall should maintain the state table to check the party's (e.g., Planning Portal's) request for additional conditions of established communication.

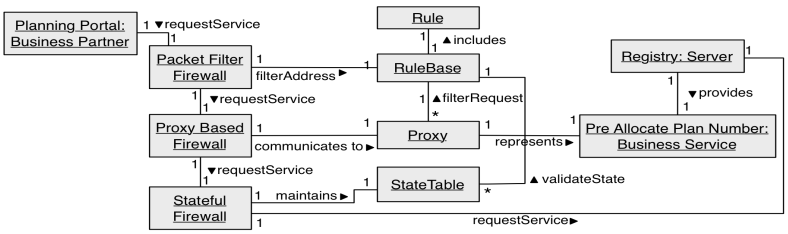

Fig. 6: Firewall Architecture, adapted from [12]

It is important to notice that the communication between the Planning Portal (and also Lodging Party) and the Registry is bidirectional. The similar requirements must be taken into account when Registry sends messages (e.g., Fee Calculation Service sends Invoice) back to the business party.

Data Store is used to define how data are stored and retrieved to/from the associated databases (e.g., Data store in Fig 2). If the threat agent is capable of accessing and retrieving the data, their confidentiality and integrity would potentially be negated, thus, resulting in the harm of the business asset (i.e., the Plan) and its supporting IS assets (i.e., the Registry).
RQ15. The server (e.g., Registry) should audit the operations after the retrieval, storage or any other manipulation of data in the data store (e.g., Data store).

Auditing is the process of monitoring and recording selected events and activities [9]. It determines who performed what operations on what data and when. This is useful to detect and trace security violations performed on the Plan Number, Digital Data and Plan Validation. Potentially, the data store auditing could be supported by the access control policy.
RQ16. The server (e.g., Registry) should perform operations to hide/unhide data when they are stored/retrieved to/from the data store (e.g., Data store).

A possible RQ16 implementation is cryptographic algorithms. The encryption offers two-fold benefits: (*i*) the data would not be seen by the Data store users (e.g., database administrator) where the circumstances do not allow one to revoke their permissions; (*ii*) due to any reasons if someone gets physical access to the Data store (s)he would not be able to see the confidential data stored.

4 Conclusion

In this paper, we presented a method to elicit security requirements from the business process models. Its strength lies in its general description of security goals and the systematic analysis of the contextual areas. In comparison to the related work where the focus is placed on representing security requirements (graphically) on the process models, our proposal suggests a novel approach to elicit these requirements and define them as the business rules.

The method should be improved with new security risk-oriented patterns. We also plan to extent the method with the requirements prioritisation to support the trade-off analysis. Finally, we continue validating the method empirically.

References

1. Accorsi, R., Stocker, T.: On the Exploitation of Process Mining for Security Audits: The Conformance Checking Case. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 1709–1716. SAC, ACM (2012)
2. Ahmed, N., Matulevičius, R.: Securing Business Processes using Security Risk-oriented Patterns. Computer Standards and Interfaces 36(4), 723–733 (2014)
3. Apostolopoulos, G., Peris, V., Saha, D.: Transport Layer Security: How Much Does It Really Cost? In: Proceedings IEEE INFOCOM'99 The Conference on Computer Communications. vol. 2, pp. 717–725 (1999)
4. Chang, R.: Defending Against Flooding-based Distributed Denial-of-Service Attacks: A Tutorial. Communications Magazine, IEEE 40(10), 42–51 (2002)
5. Clarke, J., Fowler, K., Oftedal, E., Alvarez, R.M., Hartley, D., Kornbrust, A., O'Leary-Steele, G., Revelli, A., Siddharth, S., Slaviero, M.: SQL Injection Attacks and Defense, Second Edition. Syngress Publishing, 2nd edn. (2012)
6. Dumas, M., O'Sullivan, J., Hervizadeh, M., Edmond, D., Hofstede, A.H.M.t.: Towards a Semantic Framework for Service Description. In: Semantic Issues in E-Commerce Systems. pp. 277–291. Kluwer, B.V. (2003)
7. Herrmann, P., Herrmann, G.: Security Requirement Analysis of Business Processes. Electronic Commerce Research 6(3-4), 305–335 (2006)
8. Menzel, M., Thomas, I., Meinel, C.: Security Requirements Specification in Service-Oriented Business Process Management. In: ARES. pp. 41–48 (2009)
9. Natan, R.B.: Implementing Database Security and Auditing: Includes Examples for Oracle, SQL Server, DB2 UDB, Sybase. Digital Press, Newton, MA, USA (2005)
10. Rodríguez, A., Fernández M, E., Piattini, M.: A BPMN Extension for the Modeling of Security Requirements in Business Processes. IEICE-TIS(4) pp. 745–752 (2007)
11. Röhrig, S., Knorr, K.: Security Analysis of Electronic Business Processes. Electronic Commerce Research 4(1-2), 59–81 (2004)
12. Schumacher, M., Fernandez B., E., Hybertson, D., Buschmann, F., Sommerlad, P.: Security Patterns: Integrating Security and Systems Engineering. Wiley (2006)