Bus Inf Syst Eng 60(1):3–20 (2018)
https://doi.org/10.1007/s12599-018-0518-4

RESEARCH PAPER

Security Requirements Elicitation from Airline Turnaround Processes

Raimundas Matulevičius · Alex Norta · Silver Samarütel

Received: 1 March 2017 / Accepted: 4 October 2017 / Published online: 15 January 2018

Abstract Security risk management is an important part of system development. Given that a majority of modern organizations rely heavily on information systems, security plays a big part in ensuring smooth operations of business processes. For example, many people rely on e-services offered by banks and medical establishments. Inadequate security measures in information systems have unwanted effects on an organization's reputation and on people's lives. This case study paper targets the secure system development problem by suggesting the application of security requirements elicitation from business processes (SREBP). This approach provides business analysts with means to elicit and introduce security requirements to business processes through the application of the security risk-oriented patterns (SRPs). These patterns help find security risk occurrences in business processes and present mitigations for these risks. At the same time, they reduce the efforts needed for risk analysis. In this paper, the authors report their experience to derive security requirements for mitigating security risks in the distributed airline turnaround systems.

Accepted after two revisions by the editors of the special issue.

R. Matulevičius (✉) · S. Samarütel
University of Tartu, Tartu, Estonia
e-mail: rma@ut.ee

S. Samarütel
e-mail: silver.samarytel@gmail.com

A. Norta
Tallinn University of Technology, Tallinn, Estonia
e-mail: alex.norta.phd@ieee.org

Keywords Security risk management · Security patterns · Security requirements engineering · Airline turnaround process

1 Introduction

Security is a very important quality which enables software to protect information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction (Anderson 2008). Modern organizations rely heavily on information systems, and security is essential for ensuring smooth operations of business processes. For example, airline industry with its rich socio-technical structure has experienced a quick and comprehensive adoption of information technology (Belobaba et al. 2015). A socio-technical system is a complex organizational work design where people solve problems at their workplaces with the means of sophisticated technology. In this complex environment one needs to consider arising security risks and define their countermeasures; in case of the aviation domain an underestimation of its complexity might lead to catastrophic airline crashes in the worst scenarios.

In this paper we apply the *case study* method (Runeson et al. 2012) and analyze the secure system development problem by applying an approach for security requirements elicitation from business processes (SREBP) (Ahmed 2014; Ahmed and Matulevičius 2015; Sandkuhl et al. 2015). In this extended version [the conference paper was published in Samarütel et al. (2016)], the objective is to elicit security requirements from the airline turnaround processes and, by highlighting the security risks, to show why these requirements are important.

As to the context of the research objective, communication is another critical security issue, an example being

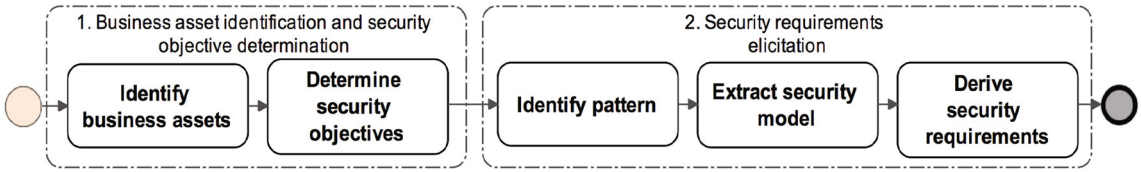

Fig. 1 The SREBP process. Adapted from Sandkuhl et al. (2015)

the deliberate jamming of automatic dependent surveillance-broadcast (ADS-B) systems (Leonardi et al. 2014), a surveillance technology to determine an aircraft position. Furthermore, we are coming to realize that the aviation industry is rapidly turning into a cyber-physical system (CPS) (Sampigethaya and Poovendran 2013) that poses additional novel risks and security issues. Briefly, a CPS (Bartelt et al. 2015) is a system composed of physical entities that are controlled or monitored by computer-based algorithms. The initial approach to studying airport-related security was rather technical while recent work recognizes this as a socio-technical system (Long 2013).

In Maiden et al. (2008), the authors recognize the socio-technical nature of airports by employing use cases and storyboards to discover stakeholder requirements such as security for the development of an airport operating system. Furthermore, in Massacci et al. (2014) the authors investigate the evolution of requirements in the context of the SecureChange[1] EU-project by means of an industry case from the Air Traffic Management (ATM) domain. Safety- and security experts are part of the focus groups while the case study results do not highlight the security specifics. Parameter measurability and social aspects of security policies in Shim et al. (2014) investigate the costs versus benefit trade-offs in alternative airport security policy constellations pertaining to, e.g., passengers or items such as baggage.

Literature shows that security-focused research for airline management is a topical area of interest. However, the topics under investigation are very specific and do not acknowledge that modern technology enables ad-hoc and process-aware collaborations (Kutvonen et al. 2012; Norta et al. 2014, 2015) which significantly reduce the amount of time and costs of airline management while yielding simultaneous improvements in service quality. Such novel ways of airline management systems also lead to unusual security risk issues for which the mitigation strategies are unclear.

The rest of the paper is structured as follows. Section 2 comprises work related to this paper. Section 3 presents the case under investigation which is about a cross-organizational airline turnaround process. Section 4 presents the results of the investigation and is followed by a discussion in Sect. 5. Finally, Sect. 6 concludes the paper and provides directions for future work.

[1] http://www.securechange.eu/.

2 Related Work

The background literature comprises two parts, namely a first part dealing with earlier studies and a second part dealing with relevant theory. For the first part about earlier studies, Sect. 2.1 introduces the method for security requirements elicitation from business processes (SREBP). Section 2.2 focuses on a set of security risk-oriented patterns and their applications, and Sect. 2.3 gives a detailed example for such a security pattern. For the second part about relevant theory, Sect. 2.4 presents means for securing business processes. Next, Sect. 2.5 discusses the elicitation and engineering of security requirements. Finally, Sect. 2.6 gives previous SREBP applications.

2.1 Security Requirements Elicitation from Business Processes

The main goal of the SREBP approach is to identify enterprise assets, to determine related security objectives, and to elicit security requirements in order to discuss and to ensure security during business-process execution (Ahmed 2014; Ahmed and Matulevičius 2015; Sandkuhl et al. 2015). Based on the guidance of the ISSRM domain model (Mayer 2009; Dubois et al. 2010), the approach integrates security into processes to enable business analysts to understand and derive the security requirements from the business-process models.

The SREBP process consists of two stages (see Fig. 1). The first stage identifies business assets and determines security objectives. It is based on the analysis of the business process models described at the different levels of abstractions, for example business value chain and business process diagrams. Specifically, the *business process diagram* expands separate actions represented in the *value chain* diagram. These diagrams describe the use of data objects, data flows and data storages (see, for example, Fig. 6). The protected *business assets*, typically, are elicited from the *value chain* and a *security objective* is determined for each identified *business asset*.

The second stage comprises as main activities (1) the identification of patterns, (2) an extraction of a security model based on pattern occurrences, and (3) a derivation of security requirements. A *security risk-oriented pattern*

(SRP) is an artifact for guiding the derivation of security risk requirements from business process diagrams. The patterns describe recurring security risks that arise within business processes. To mitigate the risks, the patterns recommend security requirements. When applying SRPs, *pattern occurrences* (i.e., a specific security context of SRP, see example in Sect. 2.3) are found in the *business process diagrams*. Pattern occurrences result in a *security model* that is extracted from the business process diagram based on the used SRP. *Security requirements* are derived from the security model.

2.2 Security Risk-Oriented Patterns

SRPs play an important role in the SREBP application. "A security pattern describes a particular recurring security problem that arises in a specific security context and presents a well-proven generic scheme for a security solution" (Schumacher et al. 2005). Based on the definition above and following the domain model for security risk management (Mayer 2009; Dubois et al. 2010), a set of security risk oriented patterns (SRPs) is suggested in Ahmed (2014) and Ahmed and Matulevičius (2014, 2015). Hence, each SRP comprises a specific security context expressed by means of asset-related concepts, as well as recurring security problems that are analyzed in terms of security risk related concepts, and suggests security countermeasures that are presented with security risk treatment concepts. Below follows a short introduction of each SRP:

- SRP1: *secures data from unauthorized access*. The security criteria is confidentiality of the data used in a business server. A user may request sensitive data from a server with the intention of misuse. To reduce the risk, the pattern proposes checking access rights. Sensitivity levels must be assigned to data- and trust levels – to people or devices accessing these data.
- SRP2: *ensures secure data transmission between business entities*. Data confidentiality and integrity are two important security criteria. However, during data transmission through a transmission medium, an interception by an attacker is possible. Thus, data could be stolen, read, changed, and (corrupted data) transmitted to the third party. In order to reduce these risks, the pattern recommends to make data unreadable and to verify data once they are received by a destination party.
- SRP3: *ensures secure business activity after data submission*. The security criteria for this pattern are availability and integrity of the business activity. Malicious scripts (e.g. SQL-, or xPath injections) submitted by means of an input interface may lead to the disruption of a business activity, rendering the latter unavailable and making it lose its integrity. Furthermore, the pattern proposes to filter incoming data, e.g., in the form of input validation, sanitation, filtration and/or canonicalization.
- SRP4: *secures business services against distributed denial of service (DDoS) attacks*. The security criterion is the availability of a business service. The risk is that a threat agent exists who creates bots of computers and sends simultaneous requests (e.g., DNS flooding, HTTP spidering, etc.) to the target server. To reduce the risk, the pattern proposes a security requirement check (i.e., filtering, classifying and detecting) for abnormal requests.
- SRP5: *secures storage of data and data retrieval from storage*. The security criterion for this pattern is confidentiality of data in the storage. The data might leak horizontally across organizational departments. A threat agent is a malicious insider with access to data in a storage. Risk reduction may involve making data invisible, or using storage monitoring and controlling.

In Sect. 2.3 we illustrate the SRP2 pattern, since it is used to show how the airline-turnaround processes are examined to determine the risks and to introduce security countermeasures.

2.3 SRP2: Ensuring Data Transmission Between Business Entities

This pattern addresses the electronic transmission of *data* between two entities (Ahmed and Matulevičius 2014, 2015), as illustrated in Fig. 2. The scenario indicates how the client fills in a form and submits data through the

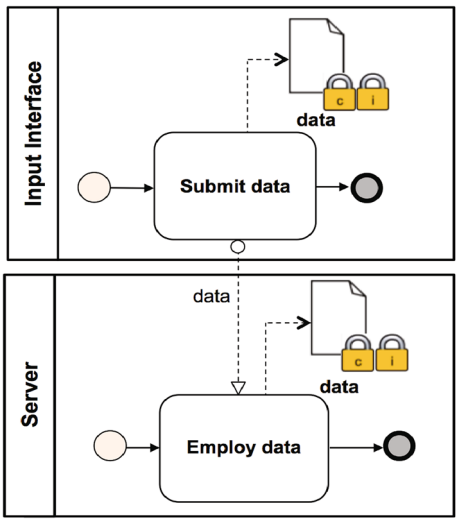

Fig. 2 SRP2: asset modeling. Adapted from Ahmed and Matulevičius (2014)

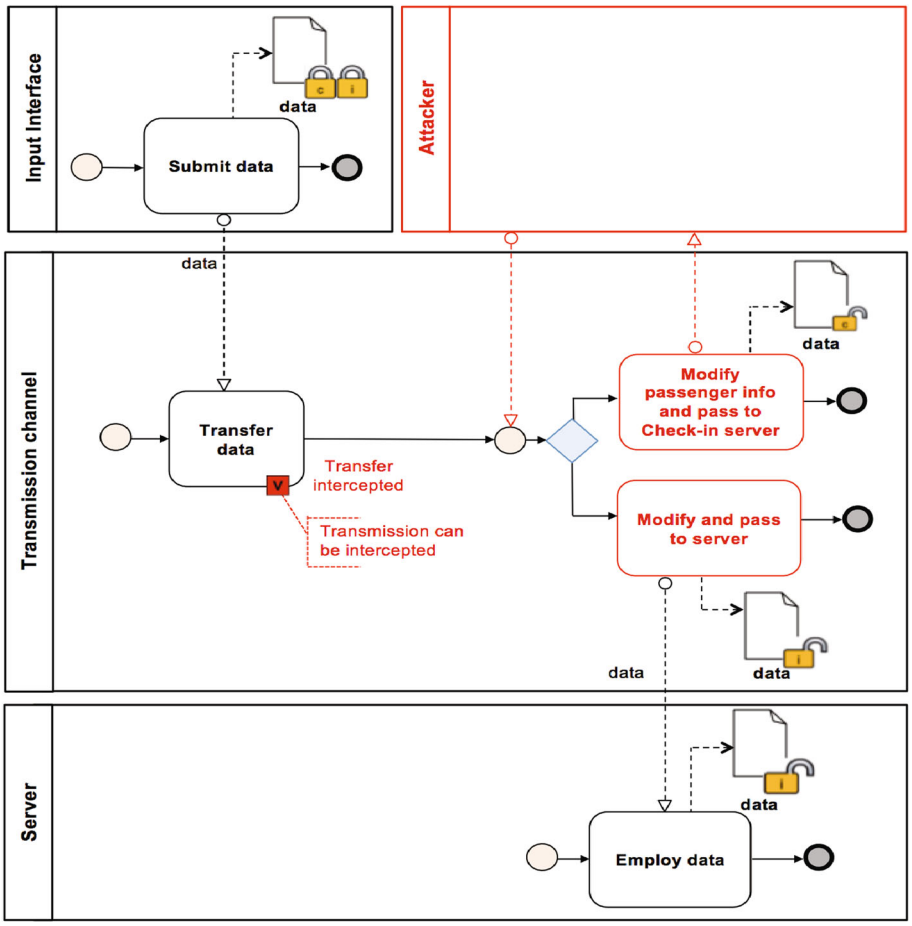

Fig. 3 SRP2: risk modeling. Adapted from Ahmed and Matulevičius (2014)

Input interface to the Server for data employment. Here, the *confidentiality* and *integrity* of data are two important security criteria.

We assume that the data are transmitted using a transmission medium (see Fig. 3). However, this situation faces (at least) two vulnerabilities. First, such a transmission medium could be intercepted by an attacker who acts as a proxy. Second, since data are not encrypted, misuse is possible, e.g., modification and passing to the server. This event harms the data, leads to the loss of transmission medium reliability, negates data integrity when data are transmitted to the server, and negates confidentiality when they are kept by the attacker.

Potential risk treatment includes risk reduction by making data unreadable and verifying the received data (see Fig. 4). The implementation includes the introduction and application of a crypto- and a checksum algorithm.

2.4 Securing Business Process

Literature suggests several approaches to enforce security on business processes. For example, Rodriguez et al. propose extensions to modeling secure business processes through understanding the security requirements (Rodriguez et al. 2007). Authors introduce non-reputation, detection of harm caused by attack, integrity, privacy, access control, security role, and security permission constructs. In Mülle et al. (2011), the security units are represented as structured text annotations tied to a particular set of the BPMN constructs (e.g., tasks, lanes, and message flows) which are equipped with the structured text annotations. The authors suggest a method to enforce the security requirements (e.g., access control, separation of duty, binding of duty and need to know principles) during the process runtime (Brucker et al. 2012).

Menzel et al. (2009) have proposed annotating the business process models with security intentions and ratings. The authors also define how to enable trustworthy interactions, organizational trust, and security intensions. The study (Schleicher et al. 2010) presents a method to impose the compliance constraints on the business processes. A concept of compliance scope is used to restrict certain areas of a business process. This helps avoiding the changes that would result in a non-compliant process. Cherdantseva et al. (2012) study how business process modeling language could be enriched with information assurance and security modeling capabilities. This happens by mapping the language constructs to the concepts of

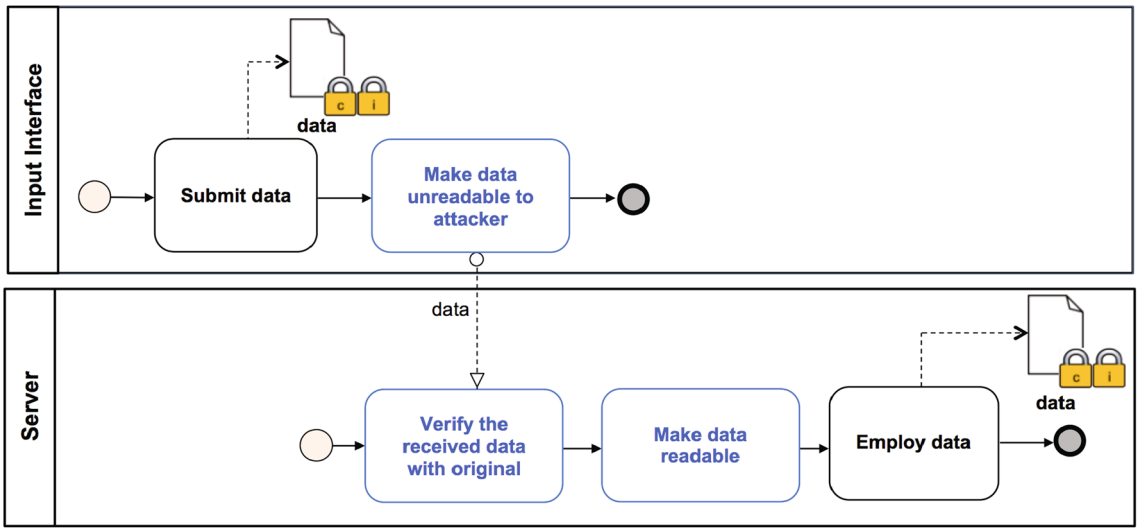

Fig. 4 SRP2: risk treatment modeling. Adapted from Ahmed and Matulevičius (2014)

information assurance and security domain model. Similarly, in Altuhhova et al. (2013) the business process model and notation are aligned to the ISSRM domain model to support security risks management in business processes. We make use of the latter extensions both to introduce SRPs in the business processes and to analyze the airline turnaround processes.

2.5 Security Requirements Elicitation and Engineering

In Fabian et al. (2010), the authors present a conceptual framework to compare and evaluate security requirements engineering approaches. Elsewhere, in Mellado et al. (2010), a systematic review is performed to classify security related approaches for techniques, frameworks, processes and methods. In this section we present a few approaches related to the SREBP in terms of the goal to elicit security requirements for the analyzed software system. For instance, in Mellado et al. (2007, 2008), the authors propose a security requirements engineering process (SREP). The approach integrates several security analysis and modeling techniques, such as common criteria (CC 2015), UMLsec (Jürjens 2005), misuse cases (Sindre and Opdahl 2005), and other.

The security quality requirements engineering (SQUARE) method (Mead and Stehney 2005; Mead et al. 2005) consists of nine steps and facilitates the use of different techniques for artifact development, risk management and assessment, security requirements elicitation and filtering. The goal for the first step is to agree on the definitions for a process. The second step is to decide upon the initial security goals. Step three involves developing, or collecting artefacts of the system being worked on. Misuse of these artefacts can be defined in misuse diagrams, goals, attack trees and other relevant models. They are important because security requirements elicitation is based on them. The fourth step is a risk assessment that consists of an assessment of the vulnerabilities and a classification of threats. The fifth step covers the selection of the security elicitation techniques. During step six, developers derive security requirements based on the outcome of previous steps. Next, two steps include security requirements categorization and prioritization. The last step is security requirements inspection. The requirements that result from earlier SQUARE steps are scrutinized to ensure that each requirement is valid and verifiable. Each of the requirements must be financially feasible for implementation.

An extension of Tropos towards security is suggested in Giorgini et al. (2005a, b) where the authors refine dependencies between actors with the concepts and visual constructs of *trust*, *delegation*, *provisioning*, and *ownership*. The *ownership* shows how the service owners access and dispose of this service. *Provisioning* describes who is allowed to provide the service. The *delegation* characterizes a formal transmission of authority by some service, e.g., from the owner to the provider. In addition authors differentiate between trust in managing permissions and trust in managing executions. Recently, this approach has been extended for socio-technical system development (Dalpiaz et al. 2016).

The mentioned security requirements elicitation and engineering approaches suggest means to elicit, document, analyze and manage security requirements. Still, they apply different techniques than suggested in the SREBP and potentially the approaches must be combined for achieving improved results.

2.6 Previous SREBP Applications

In Ahmed and Matulevičius (2015), the SREBP approach is applied in a laboratory information system and in a football federation system. The performance of the SREBP is compared to the SQUARE. In both cases it is observed that SREBP contributes with the more complete set of security requirements and, because of the predefined set of the security risk-oriented patterns, the performance of requirements derivation activities is faster. At the same time, earlier case studies do not consider distributed systems. Therefore, in this paper we focus on the SREBP approach to elicit security requirements from a distributed airline turnaround system.

3 Case Study Design

The structure of this section is as follows. First, the research questions are presented in Sect. 3.1. Next, we describe the case selection in Sect. 3.2 followed by the data-collection procedure of Sect. 3.3, the analysis- and validation procedure in Sect. 3.4.

3.1 Research Questions

For this case study, we derive the main research question of *how to apply SREBP for early stage security analysis in the airline turnaround domain* from the earlier stated research objectives in the introduction. To establish a separation of concerns, the main research question is split into the set of sub-questions below:

- **RQ1**: What are the protected assets in the airline turnaround processes?
- **RQ2**: What are security countermeasures for the airline turnaround processes?
- **RQ3**: What are the related security risks?

3.2 Case Selection

To answer these questions we illustrate the application of SREBP in the aviation-turnaround system that Nõukas (2015) first investigated and that was further explored in Matulevičius et al. (2016). The airline-turnaround process in Fig. 5 depicts three swimlanes for ground services, passenger management and gate agent respectively. The ground-services swimlane begins with a start-signal event to commence after-flight services. The following top parallel branch of the AND-split comprises a catching intermediary start signal event for all passengers being deboarded, followed by yet another AND-split for cleaning, restocking the aircraft, and fueling after a start message event indicates a fuel-slip receipt. The restocking task for the aircraft requires a passenger-information data object, e.g., comprising dietary needs. Following the AND-join, an intermediate signal event signals boarding is allowed.

The other branch of the initial AND-split commences with a cargo- and luggage offloading task, followed by an AND-split with respective intermediate message event nodes. The top parallel branch halts until it receives a message from an adjacent process indicating a cargo assignment and the second parallel branch likewise needs to wait until catching the message that the luggage receipt exists. After this the AND-join, cargo- and luggage-loading starts, culminating into another AND-join before an end-signal event terminates the process for ground services.

The middle passenger-management swimlane of Fig. 5 begins with a start timer event 24 h before the estimated time of departure (ETD). The latter follows an AND-split with the top parallel branch starting a sequence with a passenger check-in task which receives a data object with external passenger information. The task also contributes to a data object about checked-in passengers. Next, an intermediate timer event waits for 4 h before ETD for luggage check-in commences. The luggage check-in task requires the data object about checked-in passengers and produces a luggage-data object. After the completed check-in and with 1 h left until ETD, an intermediate message event sends information about the luggage being ready to the swimlane for ground services before the AND-join. The bottom parallel branch starts after-flight services with an intermediate signal event, followed by a passenger de-boarding task from the landed airplane. An intermediate signal event indicates all passengers are deplaned, followed by yet another intermediate signal event to say that boarding may proceed. The subsequent boarding-process task involves the data object for boarded passenger information with the final AND-join leading to the end signal event for signing off the preflight service.

The final gate-agent swimlane in Fig. 5 comprises a start signal event for the aircraft's arrival, followed by an intermediate signal event to start after-flight services. The gate agent monitors the turnaround process before an AND-split where in parallel two intermediate signal events indicate a preflight service sign-off for ground operation and for passenger management respectively. The following AND-join culminates in the end signal event for allowing an airplane takeoff.

3.3 Data-Collection Procedure

The airline turnaround case in Fig. 5 results from industry collaboration with an ICT-service providing company from the aviation domain. In an initial meeting with the company, the case context was presented that formerly low-

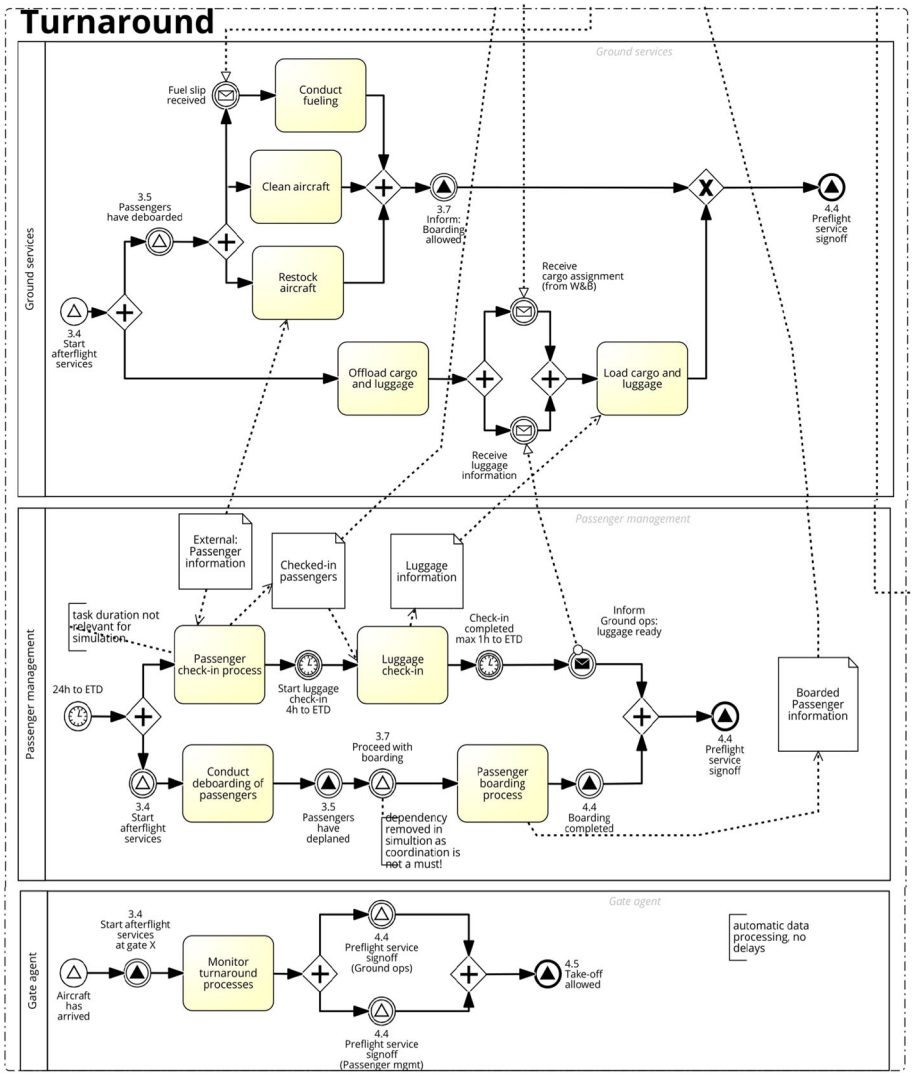

Fig. 5 Airline turnaround process. Adapted from Nõukas (2015)

tech airline-turnaround processes involving several small- and medium-sized enterprises (SME) are now significantly ICT supported and virtually integrated. Consequently, the challenges occur for aligning cross-organizationally the respective processes into an overall streamlined aviation turnaround that is additionally secured against attacks.

Given the novel and challenging context of having to align cross-organizational business-processes with a high degree of automation and the requirement of securing the resulting aviation turnaround, the data-collection happened on site with the company and its clients at a very large European airport. Over a month, workshops were conducted abroad with several employees of the aviation company and its clients in which the processes and related assets were studied and captured in models with documentation (Nõukas 2015).

After studying the turnaround processes, the aviation company initiated a follow-up project in which the focus was to study security risks and their mitigations (Matulevičius et al. 2016). Again, employees of the aviation company were interviewed over several months to collect data about security risks for the turnaround case in Fig. 5. This security-related data collection also involved interviews with customers of the aviation company.

3.4 Analysis- and Validation Procedures

As discussed in Jürjens (2005), while developing secure systems, the security engineering focus is placed on system implementation and maintenance. However, since security risk mitigation yields changes to a specification, a security analysis is important at an early phase (i.e., business-process- and requirement analysis). The benefit is the prevention of expensive design changes later in the development. In this paper, we shift the focus to the early stage of security analysis where the business processes are first captured in a conceptual and technology independent way.

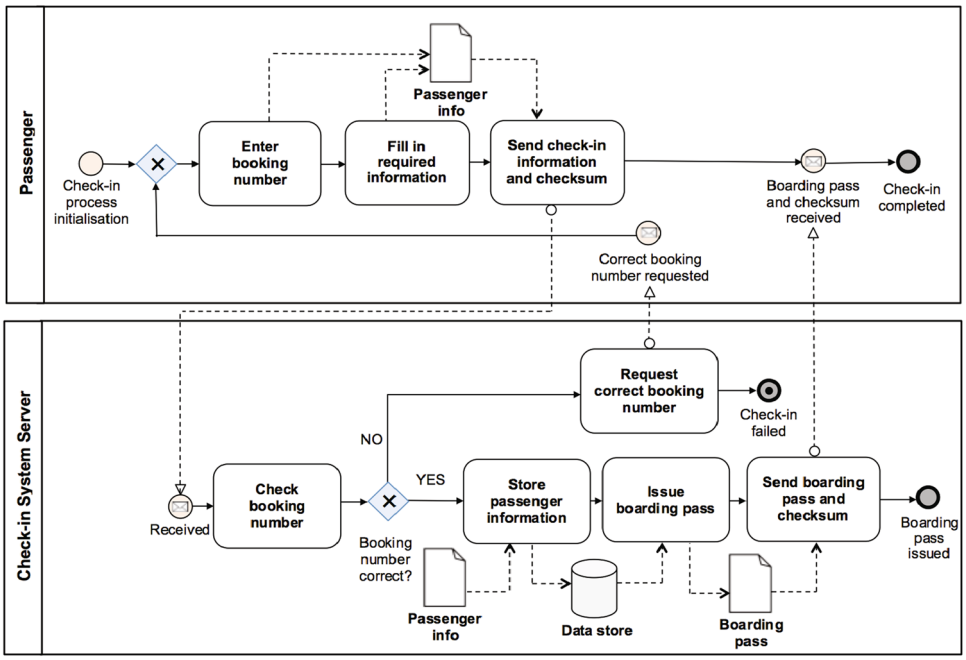

Fig. 6 Passenger check-in process

We apply the SREBP to the airline turnaround processes, reported in Nõukas (2015). The analysis scope includes five processes: (1) passenger check-in, (2) baggage check-in, (3) fuel service form issuing, (4) fuel service form requesting, and (5) loading instruction form requesting. The investigation comprises five steps:

1. *Introducing system support* The original turnaround processes, described by Nõukas (2015), include rather limited details on how the processes themselves are carried out and how they are supported by information technology systems. Hence this model presents the major value flow in the process and, thus, it can be used to understand the business assets and the security needs. However, to perform security requirements elicitation (using, e.g., SREBP), it is important to introduce how the system supports the major data exchange and data usage. The result of introducing system support is a set of models (e.g., like the one presented in Fig. 6) pertaining to the turnaround processes supported by the system.
2. *Validating models with the system expert* We have invited an expert who is knowledgeable in airline-turnaround processes to validate the developed system support process models (Samarütel 2016). The outcome of this step is a set of expert-validated models of the turnaround processes with corresponding system support.
3. *Deriving security requirements using SREBP* In this step we apply the SREBP to understand the security risks, to derive requirements and to introduce these security requirements to the analyzed processes (Samarütel 2016). The outcome of this step is the turnaround-process models enhanced with security requirements (see, e.g., Fig. 11).
4. *Performing the trade-off analysis of elicited risks and security countermeasures* The performed trade-off analysis is based on the gathered metrics for the security risk reduction, requirements cost and asset value.
5. *Validating the turnaround models enhanced with security requirements* The received process models are validated by the knowledgeable expert both for the turnaround processes and for security (Samarütel 2016). The outcome of this step is the validated turnaround-process models enhanced with security requirements.

For an extensive report about the above steps, we refer the reader to Samarütel (2016). In the next section, we report on the results of the case study steps.

4 Results

In this section, we illustrate the application of the SREBP approach to understand security requirements in the airline turnaround processes. We focus on the *Passenger check-in* process that is a part of the overall value-chain process (Nõukas 2015; Matulevičius et al. 2016) (see also Fig. 5).

By analysing the *Passenger check-in*, we answer the research questions from Sect. 3.1 while focusing on the application and usage of SRP2 and also shortly discussing the use of other patterns.

The remainder of this sections is structured as follows. Section 4.1 identifies business assets and determines security objectives. In Sect. 4.2, we use SRPs to elicit security requirements, and in Sect. 4.3 we show how patterns are used to provide the rationale for the security requirements, i.e., to visualize security risks mitigation. Finally, Sect. 4.4 reports about the application of SREBP to other turnaround processes.

4.1 Business Asset Identification and Security Objective Determination

In this section we give the answer to the first research question:

– **RQ1**: What are the protected assets in the airline turnaround processes?

Business-asset and security-objective identification is the first stage of the SREBP process (see Sect. 2) and comprises two steps: (1) identification of the business assets and (2) determination of security objectives. The key data used in the *Passenger check-in* process are related to the passenger's personal information and boarding pass. We assume that there exists a generic object that describes *Passenger Check-in details* together with attributes such as *Passenger info*, *Passenger boarding pass*, which requires protection during process execution. The following security objective comprises three security criteria and is defined as follows: (1) *integrity* of the passenger check-in details, (2) *confidentiality* of the passenger check-in details, and (3) *availability* of the passenger check-in details. This security objective is a compound element consisting of the confidentiality, integrity and availability criteria applied to the attributes of the *Passenger Check-in details*, such as *Passenger info* and *Passenger boarding pass*, and so on.

Before approaching the second research question, it is important to understand the business process diagram for the *Passenger check-in process*[2] in Fig. 6. Once the passenger initializes the process, he enters the booking number and fills in the required information (see Fill in required information), e.g., preferred seat, meal options, etc. Then the Passenger info is sent to the Check – in server. At the Check – in server the booking number is checked (see Check booking number). If the latter is not correct, the Passenger is requested to correct the check-in details (see Request correct booking number). Otherwise, the Passenger info is stored in the Data store. Next, the Boarding pass is issued (see Issue boarding pass) and sent (see Send boarding pass) to the Passenger. Once the Passenger receives the Boarding pass, the check-in process is completed.

[2] Captured using check-in process description, such as: https://www.airbaltic.com/en/online check in conditions.

4.2 Security Requirement Elicitation

Security requirement elicitation is the second step of the SREBP (see Sect. 2). By illustrating this stage we answer the second research question:

– **RQ2**: What are security countermeasures for the airline turnaround processes?

This stage includes the application of the SRPs to derive security requirements. Patterns are applied iteratively by conducting three steps: (1) identify pattern, (2) extract security model, and (3) derive security requirements. Additionally, each SRP has its own respective process for extracting a related security model as discussed in Ahmed and Matulevičius (2015). The summary of derived security requirements (and potential controls that implement these requirements) is given in Table 1. Below we illustrate how SRPs are applied. A detailed application of these patterns in the airline turnaround processes is discussed in Samarütel (2016).

4.2.1 Application of SRP1

Identify pattern No occurrences of pattern SRP1 were found in the Passenger Check – in process.

4.2.2 Application of SRP2

Identify pattern The SRP2 application derives security requirements from the *check-in process* and also introduces measures for securing the process. We identify three pattern occurrences: (1) when Passenger info is sent from Passenger to Check – in server; (2) when Check – in server requests a Passenger to deliver the correct booking number; and (3) when the Boarding pass is sent from the Check – in server to a Passenger. Below are detailed explanations for the first and third occurrences.

Extract security model To extract a security model regarding the SRP2 pattern, one needs to identify communicators and transmitted data. Regarding the first SRP2 occurrence, the extracted security model is presented in Fig. 7 where Passenger is identified as the *client* communicator and Check – in server is defined as the *server* communicator. Beforehand, Passenger info is sent while the Check – in server needs to establish secure communication details. Once the communication is established, it

Table 1 Security requirements and controls for the *Passenger check-in* process

Req. ID	Security requirements	Controls
M1.SRP2a.1	Passsenger must make passenger info unreadable for attacker before sending it to the Communication channel	Encryption algorithm
M1.SRP2a.2	Check – in server must make passenger info readable once received from the Communication channel	Encryption algorithm
M1.SRP2a.3	Check – in server must make boardingpass unreadable for attacker before sending it to the Communication channel	Encryption algorithm
M1.SRP2a.4	Passenger must make boarding pass readable once received from the Communication channel	Encryption algorithm
M1.SRP2b.1	Passenger must calculate checksum of *passenger info*	Checksum algorithm
M1.SRP2b.2	Check – in server must verify integrity of *passenger info* once received from the Communication channel	Checksum algorithm
M1.SRP2b.3	Check – in server must calculate checksum of *boarding pass*	Checksum algorithm
M1.SRP2b.4	Passenger must verify integrity of *boarding pass* when received from the Communication channel	Checksum algorithm
M1.Req3a.1	Check booking number at Check – in server must filter Passenger info when received from the communication channel	Filter input for special characters and keywords, use whitelist of acceptable inputs
M1.Req3b.1	Check booking number at Check – in server must filter confidential information from error messages and standard responses	Disable debug messages, use default error messages or error pages
M1.Req4a.1	Check – in server must filter for abnormal requests	Firewall, DoS Defence System
M1.Req5a.1	Monitor the Data store at Check – in server for malicious changes	Data access control (or Control of database signature changes)
M1.Req5b.1	Check – in server should make passenger info invisible before storing in the Data store	Encryption algorithm

is not necessary re-establish it for every data item (in other terms pattern occurrence). Thus, other data such as a request to correct booking number, boarding pass, checksum, etc., can be communicated using the established channel.

Derive security requirements After defining the security model follows the introduction of security requirements to mitigate the security risks. Figure 8 shows the security requirements to mitigate security risks identified by the first SRP2 occurrence:

- M1.SRP2a.1: a Passenger must make Passenger info unreadable to the attacker before sending it to the Communication channel.
- M1.SRP2a.2: the Check – in server must make passenger info readable once it is received from the Communication channel.
- M1.SRP2b.1: a Passenger must calculate a checksum of the passenger info.
- M1.SRP2b.2: the Check – in server must verify the integrity of the *passenger info* once received from the Communication channel.

Similar security requirements must be derived regarding the Boarding pass, as Fig. 11 shows in detail:

- M1.SRP2a.3: the Check – in server must make the boarding pass unreadable for an attacker before sending it to the Communication channel.

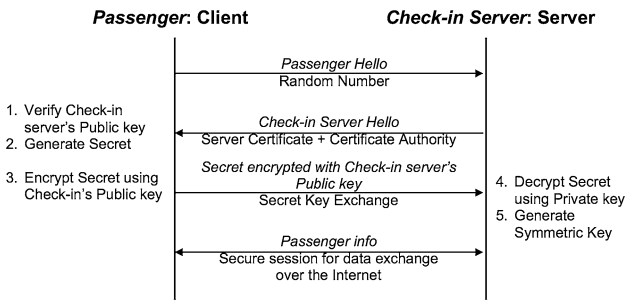

Fig. 7 SRP2: extracted security model

- M1.SRP2a.4: the Passenger must make the boarding pass readable once received from the Communication channel.
- M1.SRP2b.3: a Check – in server must calculate a checksum of the boarding pass.
- M1.SRP2b.4: the Passenger must verify the integrity of the *boarding pass* when received from the Communication channel.

Security requirements M1.SRP2a.1 – 4 are implemented using the *cryptography algorithms*; for example, see *cryptographic key management* pattern in Schumacher et al. (2005). Requirements M1.SRP2b.1 and M1.SRP2b.2 are implemented using the *checksum algorithms*.

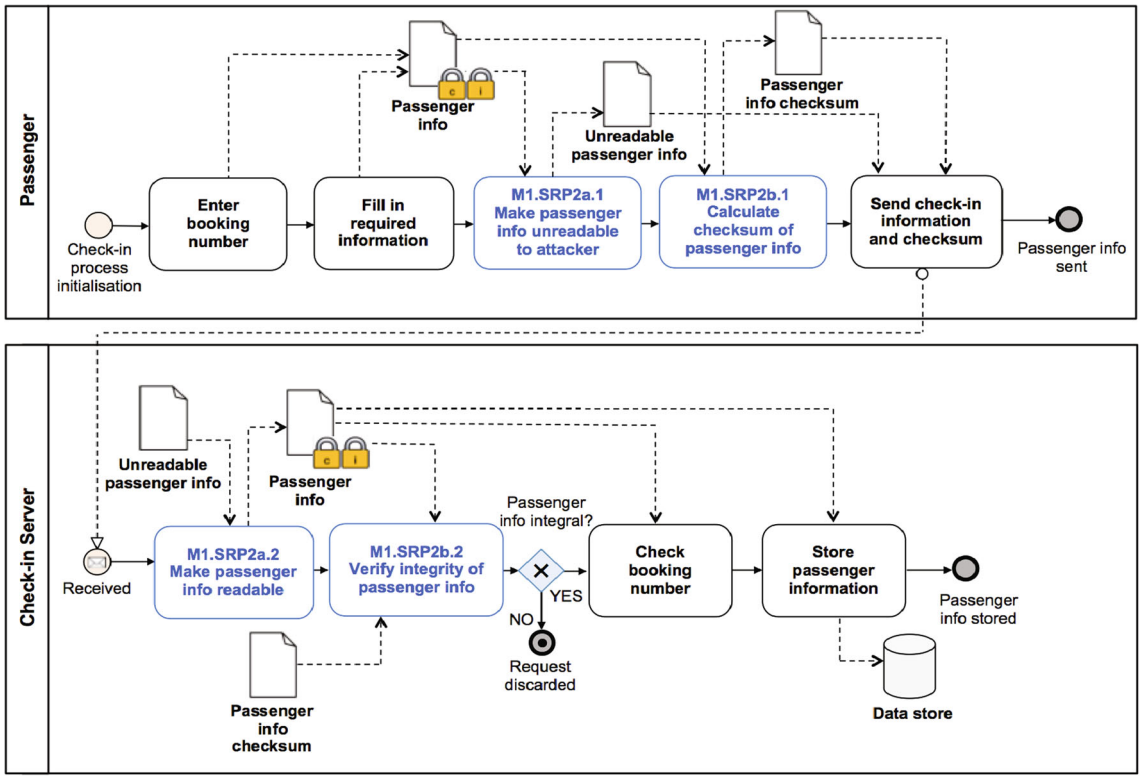

Fig. 8 Derivation of security requirements using SRP2

4.2.3 *Application of SRP3*

Identify pattern We identify one SRP3 occurrence: this is when Passenger info is sent from Passenger to Check – in server. Two other instances of communications (i.e., when a request to correct boarding number is sent from Check – in server to Passenger and when a boarding pass is sent from Check – in server to Passenger) between the Check – in server and Passenger are not relevant because the SRP3 is applied for the *input* interfaces.

Extract security model There is no graphical security model associated with SRP3. However, it is essential to determine the *input interface* (i.e., Check booking number at Check – in server) and the *input data*, i.e., Passenger info and any other data (e.g., checksum as illustrated in SRP2) submitted to the Check booking number at Check – in server.

Derive security requirements Once security-related information is defined, the following security requirements are formulated:

– M1.Req3a.1: Check booking number at Check – in server must filter Passenger info when received from the communication channel.
– M1.Req3b.1: Check booking number at Check – in server must filter confidential information from error messages and standard responses.

To strengthen these requirements and their countermeasures (Ahmed and Matulevičius 2014), security requirements for Passenger info sanitization and canonicalization (Balzarotti et al. 2008; Clarke et al. 2012) could be defined.

4.2.4 *Application of SRP4*

Identify pattern Similar to SRP3, we identify one SRP4 occurrence: when Passenger info is sent from Passenger to Check – in server.

Extract security model In order to construct a security model, one needs to identify the *functional unit*, i.e., the Check booking number at the Check – in server, and to identify the business partner, i.e., Passenger. The security model is instantiated following the network architecture model, as illustrated in Fig. 9.

Derive security requirements The following security requirement could be defined:

– M1.Req4a.1: Check – in server must filter for abnormal requests.

This "high level" requirement could be implemented by installing, for example, firewalls or DoS defence systems

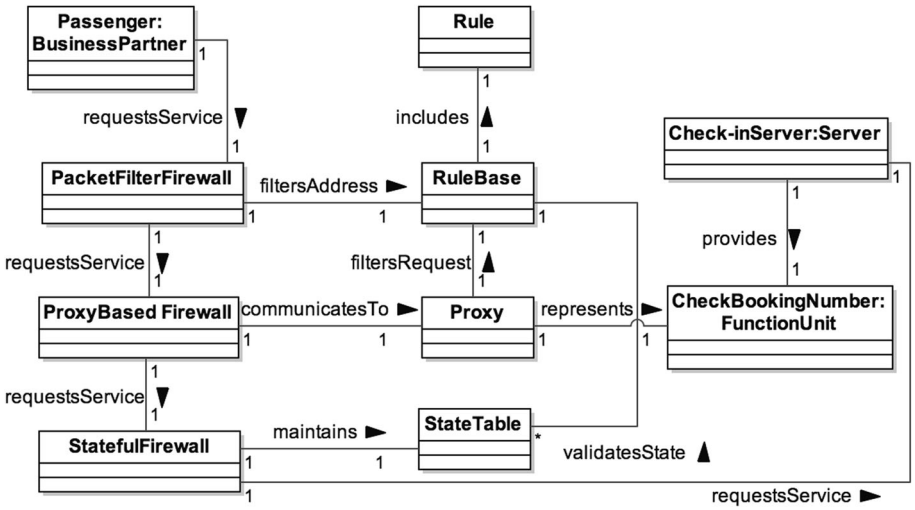

Fig. 9 SRP4: extracted security model (network architecture model adapted from Schumacher et al. 2005)

(e.g., see Fig. 9). More specifically, the concrete security requirements should be derived from the security model:

- M1.Req4a.1.1: Check – in server should block all the default incoming ports that can accept messages until ports are not explicitly opened.
- M1.Req4a.1.2: Check – in server should establish a RuleBase (i.e., constraints) to communicate with the Passenger (and other business partners).
- M1.Req4a.1.3: *Packet filter firewall* should filter the Passenger's address to determine if it is not a host used by the threat agent.
- M1.Req4a.1.4: *Proxy based firewall* should communicate to the proxy that represents the Check booking number to determine the validity of the request received from the Passenger.
- M1.Req4a.1.5: State firewall should maintain the state table to check the Passenger's request for additional conditions of established communication.

These requirements define different security levels: potentially implementing all of them guarantees the high security level. However, this influences the performance of the system (Ahmed 2014).

4.2.5 Application of SRP5

Identify pattern In principle, this pattern focuses on the data access to and from the database, or any other data storage. The pattern occurrences are found when data are loaded to (i.e., Store passenger information), and read from (i.e., Issue boarding pass) the database (i.e., Data store).

Extract security model To define the security model as in Fig. 10, it is necessary to determine the secure resources,

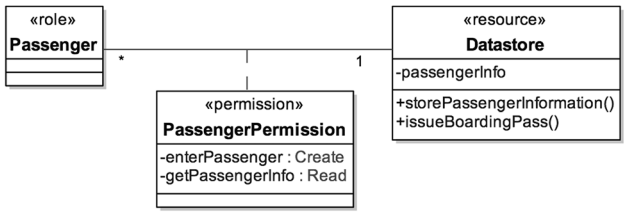

Fig. 10 SRP5: extracted security model

i.e., Datastore, operations, i.e., Store passenger information and Issue boarding pass, and secured attribute, i.e., Passenger info, which value is changed, or used when executing these operations.[3] We also assume that there is a role, i.e., Passenger, which has an interest to observe "surrounding" events or tasks, such as Check – in information received, or Send boarding pass) to load or to read the value of the attribute. A Passenger permission should be defined that describes what secure actions are allowed to be carried out on the resource.

Derive security requirements We describe a "generic" security requirement following the security model:

- M1.Req5a.1: monitor the Datastore at Check – in server for malicious changes.

This pattern infers that security constraints of the access control policy need to be established: these constraints are:

[3] In comparison to SRP1, which defines permissions to execute system activities (i.e., functions, operations), the SRP5 pattern takes into account permissions and access control constraints defined regarding the access of the data storage (e.g., database and its separate tables). Such a security model defines the access policy and contributes to the monitoring controls for the *data* access.

- M1.Req5a.1.1: only Passenger should enter (i.e., *create new entry*) passenger info to Datastore by executing operation storePassengerInformation.
- M1.Req5a.1.2: passenger info is read only by operation Issue boarding pass to issue the boarding pass to the Passenger.

In addition to the access control policy, we strengthen the security level by introducing the use of encryption algorithms. This implements the security requirement:

- M1.Req5b.1: Check – in server should make passenger info invisible before storing it in the Data store.

Figure 11 summarizes the discussion by illustrating how security requirements are introduced into the *Passenger check-in process*. Note however that alternative process designs are possible. In the next stage we asks for the rationale why these security requirements are needed in accordance with Sect. 4.3. Additionally, a trade-off analysis must be performed in order to determine, which security requirements should be implemented in the system or which security risks should be mitigated first.

4.3 Rationale for Security Requirements

In this section we consider the third research question:

- **RQ3**: What are the related security risks?

This question is not a direct target of the SREBP application. Still, for those interested in why the security countermeasures are introduced, i.e., security requirements and controls, the security models resulting from the SRP application and representing the security risks are presented to motivate the derived security countermeasures. We discuss the security risks resulting from the SRP2 application in detail and summarize other security risks identified using patterns SRP3, SRP4, and SRP5 on the *Passenger check-in process*.

Figure 12 illustrates security risk RiskID1 that is explicit if the security risk model is defined following SRP2. Here, *integrity* of the Passenger info is considered assuming that the Passenger info is sent via a Transmission channel. Consequently, an Attacker exists who is able to intercept this Transmission channel in accordance with *vulnerability* [V] – thus, Transmission can be intercepted, resulting in the *man in the middle* attack. The Attacker is able to modify passenger information and pass it on to Check – in server. This attack results in a negation of the integrity of the Passenger info in accordance with the *open lock*. At the Check – in server, the integrity of the receiving passenger info is not checked, which results in storing the changed Passenger info in the Data store.

In Fig. 13 security risk RiskID2 is illustrated. SRP2 is applied regarding the Boarding pass confidentiality. Again, the Transmission channel can be intercepted due to the same vulnerability, while this time, the Attacker reads and keeps the boarding pass (see, Read and keep boarding pass). This results in the negation of the boarding pass integrity. By acting as the *man in the middle*, the Attacker is able to change the Passenger info, e.g., by inserting his own name, and steal the Boarding pass in order to access the plane.

Other security risks that are potentially identified using SRP3, SRP4, and SRP5 are:

- RiskID3: an Attacker capable of writing malicious scripts, e.g., SQL injection, xPath injection, etc., submits malicious scripts due to the lack of the input filtering at the Check – in server, thus resulting in the loss of the integrity of the *Passenger info* and potentially the integrity of the *Issue board pass* service. The risk results from applying SRP3.
- RiskID4: an Attacker performs many simultaneous requests to the Check – in server making it not available to the Passenger, thus resulting in a loss of availability of the Issue board pass service. The risk results from applying SRP4.
- RiskID5: a (malicious) insider modifies the Passengerinfo by using the access control rights due to poor data integrity checks, thus leading to the loss of Passenger info integrity and possibly a loss of integrity or availability of the Issue board pass. The risk results from applying SRP5.

4.4 Security Requirements from Other Turnaround Processes

The SREBP approach was used to derive security requirements from other turnaround processed – *baggage check-in* (secured assets – Baggage info and Bag tags), *Fuel service form issuing* (secured assets – Fuel quantityinfo and Fuel service form), *Fuel service form requesting* (secured assets – Fuel service form request and Fuel service form), and *Loading instruction form requesting*.

Table 2 (secured assets – Loading instruction form request and Loading instruction form) summarizes the number of requirements elicited using the SRPs. The largest number of requirements we derive from the *Fuel service form requesting* process. Other analysis of the processes results in the same number of requirements. We elicit 34 security requirements using the SRP2 pattern and only 2 requirements are derived using the SRP1 pattern.

Once security requirements are derived, it is important which requirements need to be implemented. The next

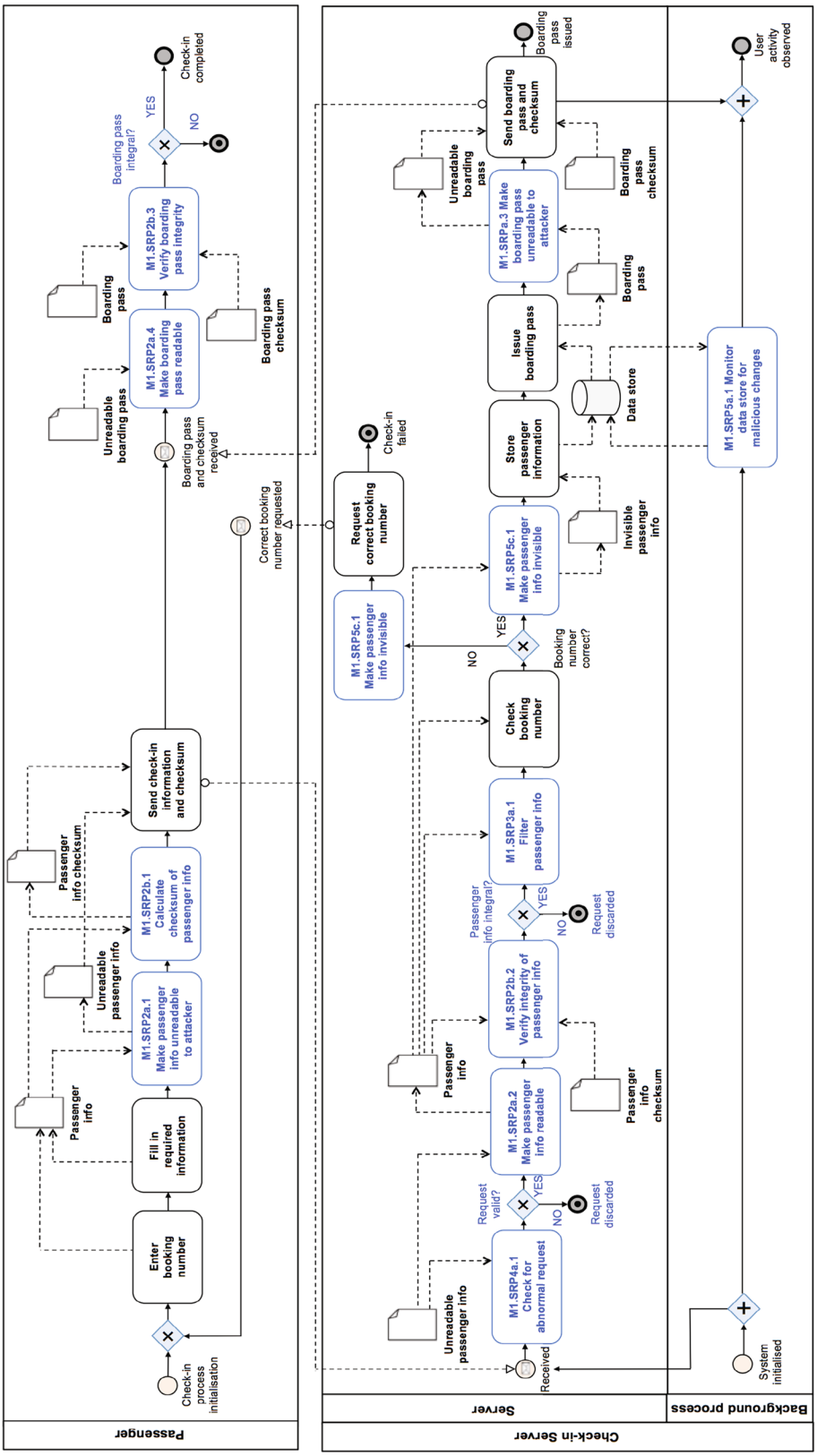

Fig. 11 Security requirements for *Passenger check-in process* derived using the security risk-oriented patterns

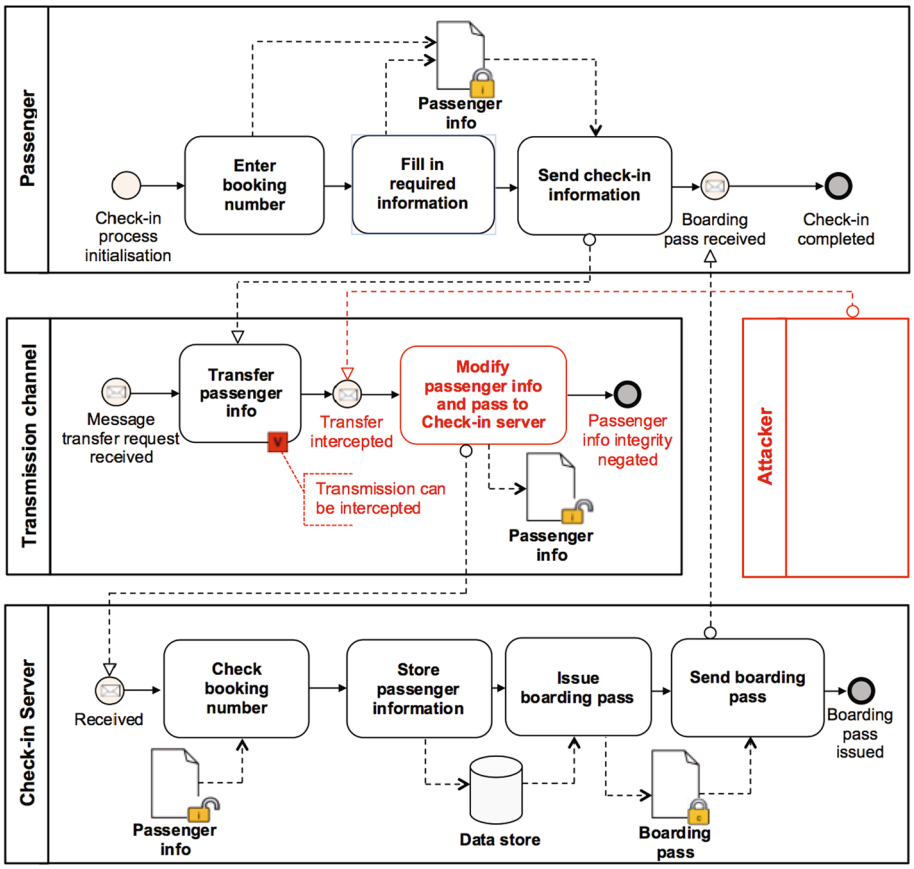

Fig. 12 Capturing potential security risks to the *Passenger info* asset

activities are requirements prioritisation and security trade-off analysis, which helps determining important security countermeasures. The implementation of the security controls in the process will certainly result in constraints to the system performance and efficiency. This analysis, however, remains outside the scope of this study.

5 Limitation and Discussion

In this paper, we employ a case study to understand security issues resulting from the collaboration between airlines and service providers. We identify relevant assets by modeling the business processes of an airline-turn-around process. We find these assets in the passenger management process and ground operations. The research result is a security requirement and control framework. The risk analysis is supported by theoretical methods from the domain of security risk management.

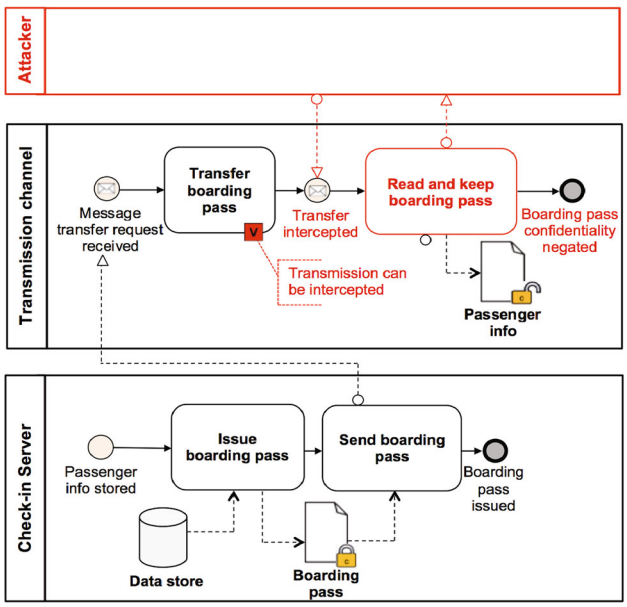

Fig. 13 Capturing potential security risk to the *Boarding pass* asset

Table 2 Number of security requirements elicited from the turnaround processes using SRPs

Processes	SRP1	SRP2	SRP3	SRP4	SRP5	Total
Passenger check-in process	–	6	2	1	2	11
Baggage check-in process	–	6	2	1	2	11
Fuel service form issuing	–	10	1	1	3	15
Fuel service form requesting	1	6	1	1	2	11
Loading instruction form requesting	1	6	1	1	2	11
Total	2	34	7	5	11	59

5.1 Study Limitation

Our analysis comprises a certain degree of subjectivity. Throughout the validation process, we only consulted one expert. Although we trust the received feedback, opinions by nature are subjective, and a collection of opinions also from other experts would be preferable.

Another limitation is that the SREBP approach is applied only to five business processes. Although the processes are based on real life scenarios, we require a larger number of process models. An interesting direction of research would be to consider business processes from other industries besides aviation to investigate how well the SREBP conform in different domains.

Finally, the SREBP approach is used only by one of its authors. Other researchers may have different observations of the business asset identification, security needs determination, requirements security patterns' applicability, and security trade-off analysis. Therefore, to potentially mitigate this risk, we would welcome feedback from practitioners and laymen who are unfamiliar with the approach and its application to business processes.

5.2 Discussion

The following observations result from the application of security risk oriented patterns:

- *The expert's feedback to the secured business processes is approving*. Revised airline turnaround models (see step *2* in Sect. 3) and security requirements (see step *5*) are approved as relevant and important by the expert. This also indicates that the used SREBP is a foundation for the future development of a security catalog pertaining to distributed systems.
- *The extent of SRP application differs for various patterns*. This observation results from the number of derived requirements. As discussed in Sect. 4, only two security requirements are derived using SRP1, i.e., access to data within the system. Additionally, 34 requirements out of 59 result completely from using SRP2, i.e., data transmission. This we explain by with the nature of the domain, i.e., we have considered a distributed system where communication plays an important role.
- *Not every SRP is applicable for distributed systems*. For instance, in Samarütel (2016) the SREBP approach is extended by several SRPs for protecting against deadlock attacks, for securing against brute force attacks, for securing against account lockout attacks, and so on. Although the listed SRPs are relevant in the business process models where such security risks are possible to capture, this is not the case in airline-turnaround processes. Consequently, SRP application depends significantly on the modeling domain and the level of model granularity.
- *The sequence of security requirements in a business process does not limit the choice between security controls*. The sequence of security requirements may vary in real-life business process models. When arranging the sequence of the security requirements in the business process models, we rely on a logical viewpoint. For example, in the *fuel service form issuing process*, we introduce that the server *verifies the integrity of fuel quantity information* before *readability access*. In reality, the implementation chosen to satisfy these requirements performs message encryption and an authentication in a reverse order. Thus, it is necessary to assure that implementers depict the business process, security requirements, and their sequence in a business process not necessarily as the end result.

One possible way to deal with the latter issue is the security trade-off analysis with the goal to highlight the severe risks and their countermeasures.

6 Conclusions and Future Work

We examined the applicability of the security requirements elicitation from business processes in five business-process models originating from airline-turnaround processes. The business processes we enhanced with security requirements using the security risk aware BNPMN modeling language. We submitted the secured business processes for review to

an expert who has experience with business processes used in the airline industry.

The case study confirms the application feasibility of the chosen approach. The study shows that there are many security issues that exist in the airline industry. Specifically problematic is that this industry segment is affected by ICT innovation at a speed where decision makers do no longer understand the evolving virtual enterprises that match their processes cross-organizationally, and are suddenly confronted with potentially catastrophic socio-technical security issues.

The implication of our results is that companies that operate in the airline industry must rapidly develop business-process awareness as a prerequisite for automation. The subsequent challenge for achieving progress in terms of operational effectiveness and efficiency is to cross-organizationally match in-house processes. While the dominant perspective explored is in this case is control flow, security issues also arise from the perspectives of data flow, resource management, exception and compensation management, and so on.

The limitation of this paper is that we can only report in this case study paper on a small set of the SRP applications for one case. Consequently, in future work we aim to expand on the study by exploring the applicability of other SREBP patterns. More specifically, we aim to study patterns that can not be applied in this airline-turnaround case study.

Acknowledgements This research was funded by the Estonia Research Council (Grant IUT20-55).

References

Ahmed N (2014) Deriving security requirements from business process models. PhD thesis, University of Tartu

Ahmed N, Matulevičius R (2014) Securing business process using security risk-oriented patterns. Comput Stand Interfaces 36:723–733

Ahmed N, Matulevičius R (2015) Presentation and validation of method for security requirements elicitation from business processes. In: Information systems engineering in complex environments, selected extended papers from CAiSE Forum 2014

Altuhhova O, Matulevičius R, Ahmed N (2013) An extension of business process model and notification for security risk management. Int J Inf Syst Model Des 4(4):93–113

Anderson R (2008) Security engineering: a guide to building dependable distributed systems, 2nd edn. Wiley, New York

Balzarotti D, Cova M, Felmetsger V, Jovanovic N, Kirda E, Kruegel C, Vigna G (2008) Saner: composing static and dynamic analysis to validate sanitization in web applications. In: Security and privacy, pp 387–401. IEEE

Bartelt C, Rausch A, Rehfeldt K (2015) Quo vadis cyber-physical systems: research areas of cyber-physical ecosystems: a position paper. In: Proceedings of the 1st international workshop on control theory for software engineering, pp 22–25, New York. ACM

Belobaba P, Odoni A, Barnhart C (2015) The global airline industry. Wiley, New York

Brucker AD, Hang I, Lückemeyer G, Ruparel R (2012) SecureBPMN: modeling and enforcing access control requirements in business processes. In: Proceedings of the 17th ACM symposium on access control models and technologies, pp 123–126. ACM

CC (2015) Common criteria for information technology security evaluation, CC v3.1. Release 4. https://www.commoncriteriaportal.org/cc/. Accessed 2 Feb 2016

Cherdantseva Y, Hilton J, Rana O (2012) Towards SecureBPMN: aligning BPMN with the information assurance and security domain. In: Business process model and notation, LNBIP, pp 107–115. Springer

Clarke J, Fowler K, Oftedal E, Alvarez RM, Hartley D, Kornbrust A, O'Leary-Steele G, Revelli A, Siddharth S, Slaviero M (2012) SQL injection attacks and defense, 2nd edn. Syngress, Burlington

Dalpiaz F, Paja E, Giorgini P (2016) Security requirements engineering: designing secure socio-technical systems. MITT Press, Cambridge

Dubois E, Heymans P, Mayer N, Matulevičius R (2010) A systematic approach to define the domain of information system security risk management. Springer, Berlin, pp 289–306

Fabian B, Gürses S, Heisel M, Santen T, Schmidt H (2010) A comparison of security requirements engineering methods. Req Eng 15(1):7–40

Giorgini P, Massacci F, Mylopoulos J, Zannone N (2005a) Modeling security requirements through ownership, permission and delegation. In: Proceedings of the 13th IEEE international conference on requirements engineering (RE'05). IEEE Computer Society

Giorgini P, Massacci F, Mylopoulos J, Zannone N (2005b) Modelling social and individual trust in requirements engineering methodologies. In: Proceedings of the 3nd international conference on trust management, LNCS, pp 161–176. Springer

Jürjens J (2005) Secure system development with UML. Springer, Heidelberg

Kutvonen L, Norta A, Ruohomaa S (2012) Inter-enterprise business transaction management in open service ecosystems. In: Enterprise distributed object computing conference (EDOC), 2012 IEEE 16th International, pp 31–40. IEEE

Leonardi M, Piracci E, Galati G (2014) ADS-B vulnerability to low cost jammers: risk assessment and possible solutions. In: Tyrrhenian international workshop on digital communications-enhanced surveillance of aircraft and vehicles, pp 41–46. IEEE

Long S (2013) Socioanalytic methods: discovering the hidden in organisations and social systems. Karnac, London

Maiden N, Ncube C, Lockerbie J (2008) Inventing requirements: experiences with an airport operations system. In: Paech B, Rolland C (eds) Proceedings of REFSQ 2008. Springer, Heidelberg, pp 58–72

Massacci F, Paci F, Tedeschi A (2014) Assessing a requirements evolution approach: empirical studies in the air traffic management domain. J Syst Softw 95:70–88

Matulevičius R, Norta A, Udokwu C, Nõukas R (2016) Security risk management in the aviation turnaround sector. In: Proceeding of FDSE 2016, pp 119–140

Mayer N (2009) Model-based management of information system security risk. PhD thesis, University of Namur

Mead NR, Stehney T (2005) Security quality requirements engineering (SQUARE) methodology. In: Software Engineering for Secure Systems (SESS05)

Mead NR, Hough ED, Stehney II TR (2005) Security quality requirements engineering (SQUARE) methodology. Technical

Report CMU/SEI-2005-TR-009, ESC-TR-2005-009, Software Engineering Institute

Mellado D, Fernández-Medina E, Piattini M (2007) A common criteria based security requirements engineering process for the development of secure information systems. Comput Stand Interfaces 29(2):244–253

Mellado D, Fernández-Medina E, Piattini M (2008) Towards security requirements management for software product lines: a security domain requirements engineering process. Comput Stand Interfaces 30(6):361–371

Mellado D, Blanco C, Sánchez LE, Fernández-Medina E (2010a) A systematic review of security requirements engineering. Comput Stand Interfaces 32:153–165

Menzel M, Thomas I, Meinel C (2009) Security requirements specification in service-oriented business process management. In: International conference on availability, reliability and security, pp 41–49

Mülle J, Stackelberg S, Bohm K (2011) A security language for BPMN process models. Technical Report 9, Karlsruhe Reports in Informatics

Nõukas R (2015) Service brokering environment for an airline. Master's thesis, Tallinn University of Technology

Norta A, Grefen P, Narendra NC (2014) A reference architecture for managing dynamic inter-organizational business processes. Data Knowl Eng 91:52–89

Norta A, Ma L, Duan Y, Rull A, Kõlvart M, Taveter K (2015) eContractual choreography-language properties towards cross-organizational business collaboration. J Int Serv Appl 6(1):1–23

Rodriguez A, Fernandez-Medina E, Piattini M (2007) A BPMN extension for the modeling of security requirements in business processes. IEICE Trans Inf Syst 90(4):745–752

Runeson P, Höst M, Rainer A, Regnell B (2012) Case study research in software engineering: guidelines and examples. Wiley, New York

Samarütel S (2016) Revision of security risk-oriented patterns for distributed systems. Master's thesis, University of Tartu

Samarütel S, Matulevičius R, Norta A, Nõukas R (2016) Securing airline-turnaround processes using security risk-oriented patterns. In Proceedings of PoEM 2016, pp 209–224

Sampigethaya K, Poovendran R (2013) Aviation cyber-physical systems: foundations for future aircraft and air transport. Proc IEEE 101(8):1834–1855

Sandkuhl K, Matulevičius R, Ahmed N, Kirikova M (2015) Refining security requirement elicitation from business process using method engineering. In: Joint proceedings of the BIR 2015 workshops and doctoral consortium

Schleicher D, Leymann F, Schumm D, Weidmann M (2010) Compliance scopes: extending the BPMN 2.0 meta model to specify compliance requirements. In: IEEE international conference on service-oriented computing and applications, pp 1–8. IEEE

Schumacher M, Fernandez E, Hybertson D, Buschmann F (2005) Security patterns: integrating security and systems engineering. Wiley, New York

Shim W, Massacci F, Tedeschi A, Pollini A (2014) A relative cost-benefit approach for evaluating alternative airport security policies. In: 9th international conference on availability, reliability and security, pp 514–522. IEEE

Sindre G, Opdahl AL (2005) Eliciting security requirements with misuse cases. Requir Eng J 10(1):34–44