Securing Airline-Turnaround Processes Using Security Risk-Oriented Patterns

Silver Samarütel, Raimundas Matulevičius, Alex Norta, Rein Nõukas

Securing Airline-Turnaround Processes Using Security Risk-oriented Patterns

Silver Samarütel[1], Raimundas Matulevičius[1], Alex Norta[2], and Rein Nõukas[2]

[1] University of Tartu, Tartu, Estonia
silver.samarytel@gmail.com and rma@ut.ee
[2] Tallinn University of Technology, Tallinn, Estonia
alex.norta.phd@ieee.org and rein.noukas@gmail.com

Abstract. Security risk management is an important part of system development. Given that a majority of modern organisations 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 organisation's reputation and on people's lives. In this case study research paper, we target the secure system development problem by suggesting the application of security risk oriented patterns. These patterns help find security risk occurrences in business processes and present mitigations for these risks. They provide business analysts with means to elicit and introduce security requirements to business processes. At the same time, they reduce the efforts needed for risk analysis. These security risk oriented patterns are applied on business processes from an aviation-turnaround system. In this paper, we report our experience to derive security requirements to mitigate security risks in distributed systems.

1 Introduction

Security is a very important software quality for the ability to protect information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction [3]. Modern organisations rely heavily on information systems and security is essential for ensuring smooth operations of business processes. For example, the socio-technically rich case of airline industry experiences a quick and holistic penetration with information technology [5]. A socio-technical system is a complex organizational work design in which people solve problems at their workplaces with the means of rather sophisticated technology. This trend leads to many new risks and security issues that are associated with civil aviation resulting in worst cases of catastrophic airline crashes. Communication is another critical security issue, e.g., a deliberate jamming of automatic dependent surveillance-broadcast (ADS-B) systems [9], a surveillance technology to determine an aircraft position. Furthermore, the recognition arises that the aviation industry turns rapidly into a cyber-physical system (CPS) [18] that poses additional novel risks and security issues. Briefly, a

CPS [4] is a system composed of physical entities that are controlled or monitored by computer-based algorithms. The initial approach to studying airport-related security is rather technical while recent work recognises this is a socio-technical system [10].

In [11], 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 [12] the authors investigate requirements evolution in the context of the SecureChange[3] EU-project with 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 explicitly address security specifics. Parameter measurability and social aspects of security policies in [20], investigate the costs versus benefit trade-offs in alternative airport security policy constellations pertaining to, e.g., passengers, items such as baggage, and so on.

Literature shows security-focused research for airline management is a topical area of interest. But the topics under investigation are very specific and do not acknowledge modern technology enables ad-hoc and process-aware collaborations [8] [15] [16] that benefit significantly the reduction of time and costs of airline management while yielding simultaneously 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.

In this case study research paper we target the secure system development problem by suggesting an application of security risk oriented patterns [1, 2]. These patterns help find security risk occurrences in business processes and present mitigations for these security risks. They provide business analysts with means to elicit and introduce security requirements to business processes. More specifically, we consider how security risk oriented patterns could be used in distributed systems, such as an aviation-turnaround system [14]. Consequently, we pursue the research objective to understand *the applicability of specific security risk oriented patterns (SRPs) that have the purpose of securing business processes in distributed systems*. More explicitly, in this paper we report our experience in applying the security risk oriented patterns in aviation-turnaround business processes.

The rest of the paper is structured as follows. Section 3 comprises related work for this paper and Section 4 presents the case under investigation about a cross-organisational airline turnaround process. Section 5 gives the results of the investigation that is followed by a discussion in Section 6. Finally, Section 7 concludes the paper and provides directions for future work.

2 Related Work

"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

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

security solution" [19]. Software projects tend to run into similar problems. Often these problems do not require new tailor-made solutions, but can be solved with solutions that have already been successfully applied in previous situations. This is where patterns come in handy. Instead of spending time and resources on working out new solutions, software developers can opt to implement already proven solutions by applying the appropriate patterns. Patterns are not independent islands. They are part of a hierarchy where larger patterns contain smaller patterns that solve sub-problems of the main problem. Patterns can be combined together with other patterns and form a larger design. Because of this combinability, patterns can effectively be applied in complex and large scale distributed systems.

There exist numerous classification systems for categorising security patterns. For instance, in [19] Schumacher *et al* presents a taxonomy comprising enterprise security and risk management, identification and authentication, access control, accounting, firewall, crypto-key management and other security pattern classes. We are also aware of numerous resources available for threat patterns (e.g., CAPEC[4], STRIDE [21], and a security threat taxonomy for distributed systems [22]. In this paper, we focus on the SRPs [1] [2] that help determining security requirements from the business processes.

3 Security Risk-Oriented Patterns

A set of security risk oriented patterns (SRPs) is suggested in [1] [2]. They are developed using a domain model [6] [13] for information system security risk management (ISSRM). This domain model differentiates between three major concept groups – *asset-related concepts*, *risk-related concepts* and *risk treatment-related concepts*. Thus, based on this structure, each SRP comprises a specific security context expressed with asset-related concepts, recurring security problem (analysed in terms of security risk related concepts) and suggests security countermeasures that are presented with security risk treatment concepts.

3.1 Patterns Used in this Study

Below, we shortly characterise each SPR used in this study:

- SRP1: *secures data from unauthorised access*. The security criteria is confidentiality of the data used in a business server. A user might request sensitive data from the server with the intention of misusing it. 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, data transmitted through a transmission medium could be intercepted by an attacker. Thus, the data could be stolen, read, changed, and transmitted to

[4] https://capec.mitre.org

the party. In order to reduce these risks, the pattern recommends to make data unreadable and to verify data once they are received at the 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 through an input interface could lead to a disruption of the business activity, making the business activity unavailable and lose its integrity. Furthermore, the pattern proposes a filtering of incoming data, e.g., in the form of input validation, sanitation, filtration and/or canonicalisation.
- 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 there exists a threat agent who creates bots of computers and runs simultaneous requests (e.g., DNS flooding, HTTP spidering, etc.) at the target server. To reduce the risk, the pattern proposes a security requirement checking (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 at the storage. The data might leak horizontally across the organisation's departments. A threat agent is a malicious insider with access to data in a storage. Risk could be reduced by making data invisible or using storage monitoring and controlling.

In Section 3.2 we discuss the SRP2 pattern, since it is used to illustrate the analysis of the airline-turnaround processes.

3.2 SRP2: Ensuring Data Transmission Between Business Entities

This pattern addresses the electronic transmission of *data* between two entities, as illustrated in Figure 1. The scenario indicates how the client fills in a form and submits data through the Input interface to the Server for data employment. Here, the *confidentiality* and *integrity* of data are two important security criteria.

The assumption is made that the data are transmitted using Transmission medium (see Figure 2). However, this situation faces (at least) two vulnerabilities. Firstly, such a transmission medium could be intercepted by an Attacker who acts as a proxy. Secondly, since data are not encrypted, they could be misused,

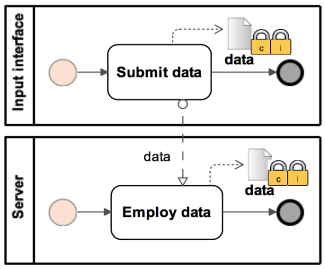

Fig. 1. SRP2: asset modeling.

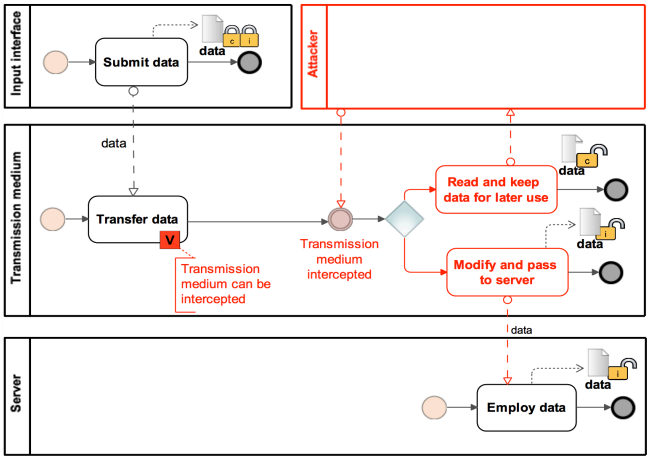

Fig. 2. SRP2: risk modeling.

e.g., modified and passed to the Server. This event harms the data, leads to the loss of transmission medium reliability, and negates data integrity (if data are transmitted to the server) and confidentiality (if they are kept by the attacker).

Potential risk treatment includes risk reduction by making data unreadable and verifying the received data (see, Figure 3). The implementation includes the introduction and application of a crypto- and a checksum algorithms.

4 Study Design

As discussed in [7], 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, 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.

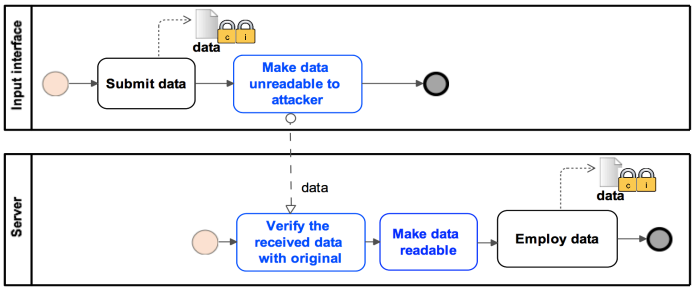

Fig. 3. SRP2: risk treatment modeling.

In this paper, we shift the focus to the early stage of security analysis where first the business processes are captured in a conceptual and technology independent way. Consequently, we pose the main research question of *how to apply SRPs for early stage security analysis in the airline turnaround domain.* To establish a separation of concerns and manageable complexity, we deduce the sub-questions. What is the appropriate case study design for exploring the suitability of the security risk oriented patterns? What analysis approach finds risks in the airline-turnaround case? What validity does the case study analysis have?

We apply the five SRPs to the airline turnaround processes, reported in [14]. The analysis scope includes five processes: (*i*) passenger check-in, (*ii*) baggage check-in, (*iii*) fuel service form issuing, (*iv*) fuel service form requesting, and (*v*) loading instruction form requesting. The investigation comprises four steps:

1. *Introducing system support*: The original turnaround processes, described by Nõukas in [14], include rather limited details on how the processes themselves are carried out and how they are supported by information technology systems. The first step is to introduce and model system support by illustrating the major data exchange and usage. The result of this step is a set of models 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. The outcome of this step is expert-validated models of the turnaround processes with corresponding system support.
3. *Deriving security requirements using patterns*: In this step we apply the SRPs to understand the security risks, to derive requirements and to introduce these security requirements to the analyzed processes. The outcome of this step is the turnaround-process models enhanced with security requirements.
4. *Validating the turnaround models enhanced with security requirements*: The received process models are validated by the expert knowledgeable both in the turnaround processes and in security. 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 [17]. In next section, we report on the results of the above steps.

5 Analyzing Airline-Turnaround Processes

First, we present the passenger check-in process, followed by an illustration of how the SPR2 pattern is applied. Next, we summarise the derived security requirements. Finally, we discuss output of other pattern applications.

5.1 Passenger Check-in Process

Figure 4 represents process for passenger check-in[5]. Once the Passenger initialises the process, he enters the booking number and fills in the required information

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

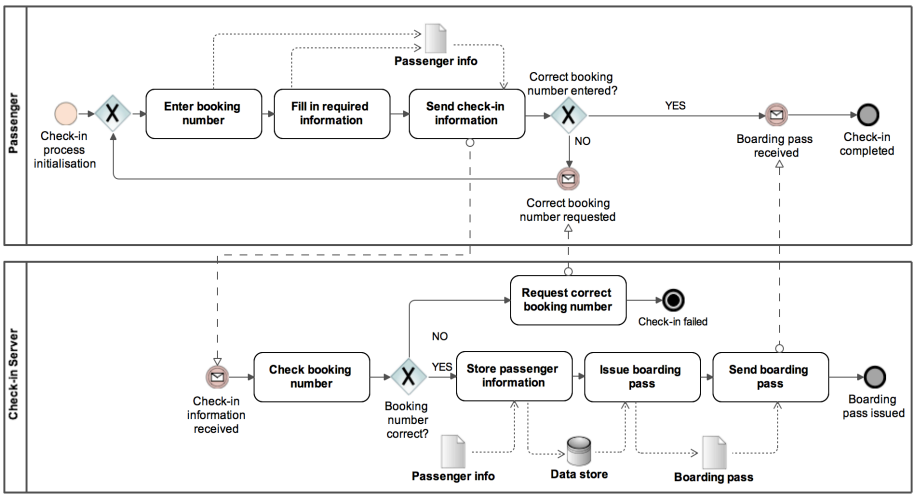

Fig. 4. Passenger check-in process.

(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 it 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.

5.2 Application of the SRP2 Pattern

We illustrate how SRP2 is applied to derive security requirements from the *check-in process* and we also introduce measures for securing the process. In the given case, we identify three pattern occurrences: (*i*) when Passenger info is sent from Passenger to Check-in Server; (*ii*) when Check-in Server requests Passenger for the correct booking number; and (*iii*) when Boarding pass is sent from Check-in Server to Passenger. In the given example, we specifically will focus on the first and third occurrences.

In Figure 5 we consider *integrity* of the Passenger info assuming that the Passenger info is sent using a Transmission channel. However, there exists an Attacker who is able to intercept this Transmission channel (see, *vulnerability* [V] – Transmission can be intercepted), thus resulting in the *man in the middle* attack. The Attacker is able to modify passenger information and pass to Check-in Server. This attack results in a negation of integrity of the Passenger info (see the *open lock*). At the Check-in Server, the integrity of the receive passenger info is not checked, which results in storing the changed Passenger info to the Data store.

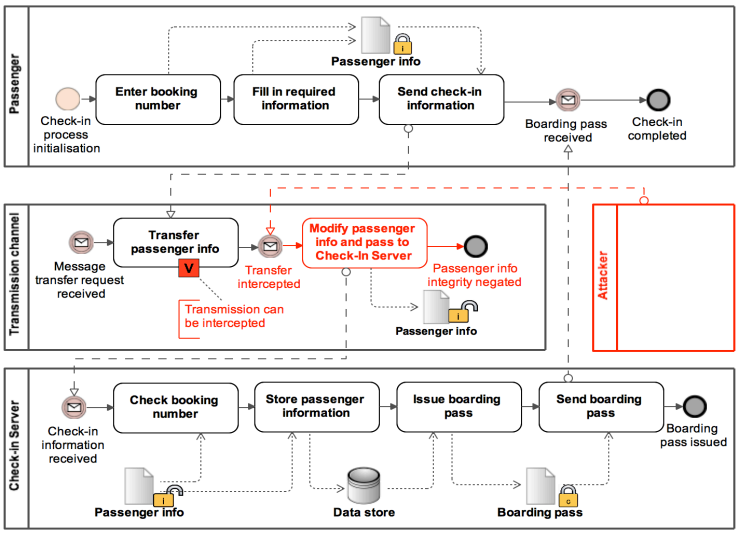

Fig. 5. Capturing potential security risks to the *Passenger info* asset.

In Figure 6, SRP2 is applied regarding the Boarding pass confidentiality. Again, the Transmission channel can be intercepted due to the same vulnerability. But 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.

5.3 Risk Treatment

To mitigate the first risk, Figure 7 shows the following security requirements are derived using the SRP2 pattern:

- M1.SRP2a.1: A Passenger should 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 should 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 Figure 8 shows in detail:

- M1.SRP2a.3: The Check-in Server should make the boarding pass unreadable to an attacker before sending it to the Communication channel.

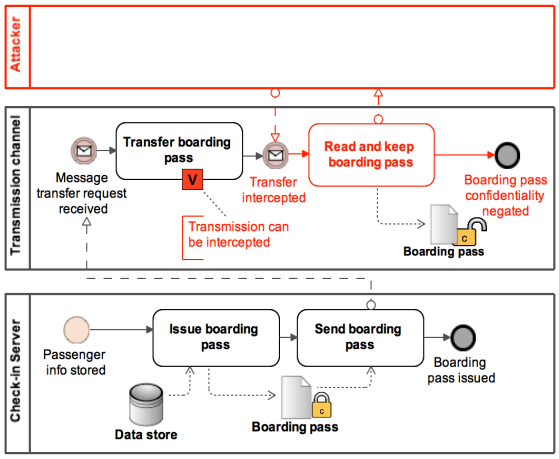

Fig. 6. Capturing potential security risk to the *Boarding pass* asset.

- M1.SRP2a.4: The Passenger must make the boarding pass readable once received from the Communication channel.
- M1.SRP2b.3: A Check-in Server should calculate a checksum of the boarding pass.
- M1.SRP2b.4: The Passenger must verify the integrity of the *boarding pass* once 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 [19]. Requirements M1.SRP2b.1 and M1.SRP2b.2 are implemented using the *checksum algorithms*.

5.4 Other Patterns

Application of patterns SRP3, SRP4, and SRP5 to the *Passenger Check-in Process* results in at least the following security risks:

- 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 integrity of the *Issue board pass* service. The risk results from applying SRP3.
- 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.
- A (malicious) insider modifies Passenger info by using the access control rights due to the poor data integrity checks, thus leading to the loss of Passenger info integrity and possibly loss of integrity and/or availability of the Issue board pass. The risk results from applying SRP5.

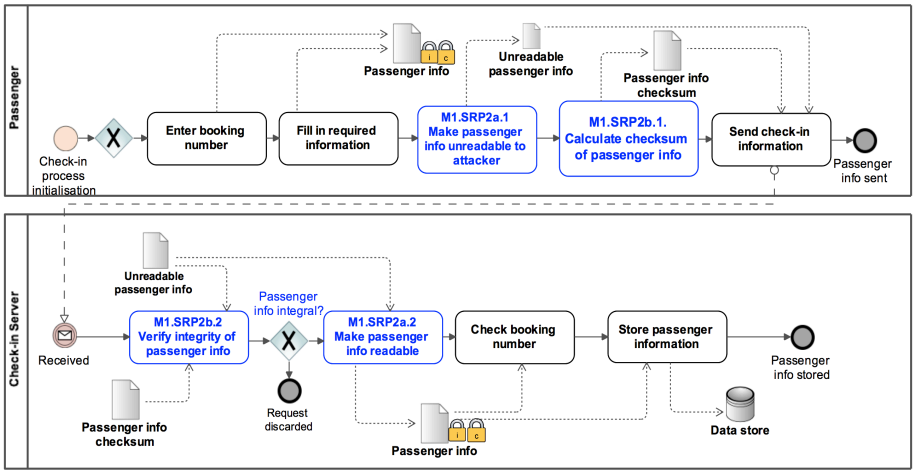

Fig. 7. Derivation of security requirements using SPR2.

To mitigate these risks security requirements are introduced in Figure 8. These security requirements are derived using security-risk-oriented patterns. Their potential implementations, i.e., security controls, are listed in Table 1.

5.5 Application of SRPs to Other Turnaround Processes

The security risk oriented patterns we apply 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 quantity info 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) summarises 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 we derive using the SRP1 pattern.

5.6 Study Limitation

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

Another limitation is that we apply the security patterns 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 is to apply

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 to 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 boarding pass unreadable to 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* once received from the Communication channel.	Checksum algorithm
M1.Req3a.1	Check-in Server must filter passenger input once received from the Communication channel.	Filter input for special characters and keywords, use whitelist of acceptable inputs
M1.Req3b.1	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	Control database signature changes
M1.Req5c.1	Check-in Server should make passenger info invisible before storing in the Data store	Encryption algorithm, monitor data access

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**

the security patterns to business processes from other industries besides aviation to investigate how well they conform in a different domains.

What should also be considered is that the security patterns are applied to the example business processes by the author of the patterns. Other researchers may have different observations of the security patterns' applicability. We request feedback from practitioners and laymen who are unfamiliar with the SRPs and apply them to business processes.

6 Discusssion

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 modelling the business processes of an airline-turnaround 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 theoretical methods from the domain of security risk management.

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 Section 4) and security requirements (see step *4*) are approved as relevant and important by the expert. This also indicates that the applied SRPs are a foundation for the future development of a security catalog pertaining to distributed systems.
- *The SRP application extent is different for various patterns.* This observation results from the number of derived requirements. As discussed in Section 5, only two security requirements are derived using SRP1, i.e., access to data within the system. Additionally, 34 requirements out of 59 result in total from using SRP2, i.e., data transmission. This we explaine with the nature of the domain, i.e., a distributed system where communication plays an important role.
- *Not every SRP is applicable for the distributed systems.* For instance, in [17], few other SRPs are suggested. For example, the SRP for protecting against deadlock attacks, the SRP for securing against brute force attacks, the SRP for securing against account lockout attacks, and few other. Although these SRPs are relevant in the business process models where these security risks are possible to capture, this is not the case in the airline-turnaround processes. This again indicates that SRP application very much depends 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 these business process, security requirements, and their sequence in the business process not necessarily as the end result.

7 Conclusions and Future Work

We examine the applicability of the security-risk oriented patterns in five business-process models originating from airline-turnaround processes. The business processes we enhance with security requirements derived from the security patterns using the security risk aware BPMN modelling language. We submit the secured business processes for review to an expert who has experience with business processes used in the airline industry.

As relations to existing evidence, the case study confirms the application feasibility of the chosen patterns. 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 not understand the evolving virtual enterprises that match their processes cross-organizationally 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 explored perspective 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 only can report on a very limited pattern application for one case due to page limitation. Consequently, in future work we aim to expand on the study by exploring the applicability of other patterns. More specifically, we aim to study patterns that did not apply in this airline-turnaround case study.

References

1. Ahmed, N., Matulevičius, R.: Securing Business Process Using Security Risk-oriented Patterns. Computer Standards and Interfaces 36, 723 – 733 (2014)
2. Ahmed, N., Matulevičius, R.: 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 (2015)
3. Anderson, R.: Security Engineering: A Guide to Building Dependable Distributed Systems, 2nd edition. Wiley (2008)

4. Bartelt, C., Rausch, A., Rehfeldt, K.: 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. CTSE 2015, ACM, New York, NY, USA (2015)
5. Belobaba, P., Odoni, A., Barnhart, C.: The global airline industry. John Wiley & Sons (2015)
6. Dubois, E., Heymans, P., Mayer, N., Matulevičius, R.: A Systematic Approach to Define the Domain of Information System Security Risk Management, pp. 289–306. Springer (2010)
7. Jürjens, J.: Secure System Development with UML. Springer-Verlag Berlin Heidelberg (2005)
8. Kutvonen, L., Norta, A., Ruohomaa, S.: Inter-enterprise business transaction management in open service ecosystems. In: Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International. pp. 31–40. IEEE (2012)
9. Leonardi, M., Piracci, E., Galati, G.: Ads-b vulnerability to low cost jammers: Risk assessment and possible solutions. In: Digital Communications-Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV), 2014 Tyrrhenian International Workshop on. pp. 41–46. IEEE (2014)
10. Long, S.: Socioanalytic methods: discovering the hidden in organisations and social systems. Karnac Books (2013)
11. Maiden, N., Ncube, C., Lockerbie, J.: Inventing Requirements: Experiences with an Airport Operations System. In: Paech, B., Rolland, C. (eds.) Proceedings of REFSQ 2008. pp. 58–72. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
12. Massacci, F., Paci, F., Tedeschi, A.: Assessing a requirements evolution approach: Empirical studies in the air traffic management domain. Journal of Systems and Software 95, 70–88 (2014)
13. Mayer, N.: Model-based Management of Information System Security Risk. Phd thesis, University of Namur (2009)
14. Nõukas, R.: Service Brokering Environment for an Airline. Master's thesis, Tallinn University of Technology (2015)
15. Norta, A., Grefen, P., Narendra, N.: A reference architecture for managing dynamic inter-organizational business processes. Data & Knowledge Engineering 91(0), 52 – 89 (2014)
16. Norta, A., Ma, L., Duan, Y., Rull, A., Kõlvart, M., Taveter, K.: eContractual choreography-language properties towards cross-organizational business collaboration. Journal of Internet Services and Applications 6(1), 1–23 (2015)
17. Samarütel, S.: Revision of Security Risk-oriented Patterns for Distributed Systems. Master's thesis, University of Tartu (2016)
18. Sampigethaya, K., Poovendran, R.: Aviation cyber–physical systems: foundations for future aircraft and air transport. Proceedings of the IEEE 101(8), 1834–1855 (2013)
19. Schumacher, M., Fernandez, E., Hybertson, D., Buschmann, F.: Security Patterns: Integrating Security and Systems Engineering. John Wiley and Sons (2005)
20. Shim, W., Massacci, F., Tedeschi, A., Pollini, A.: A relative cost-benefit approach for evaluating alternative airport security policies. In: Availability, Reliability and Security (ARES), 2014 Ninth International Conference on. pp. 514–522. IEEE (2014)
21. Shostack, A.: Threat Modeling: Designing for Security. John Wiley and Sons (2014)
22. Uzunov, A.V., Fernandez, E.B.: An Extensible Pattern-based Library and Taxonomy of Security Threats for Distributed Systems. Computer Standards and Interfaces 36(4), 734 – 747 (2013)

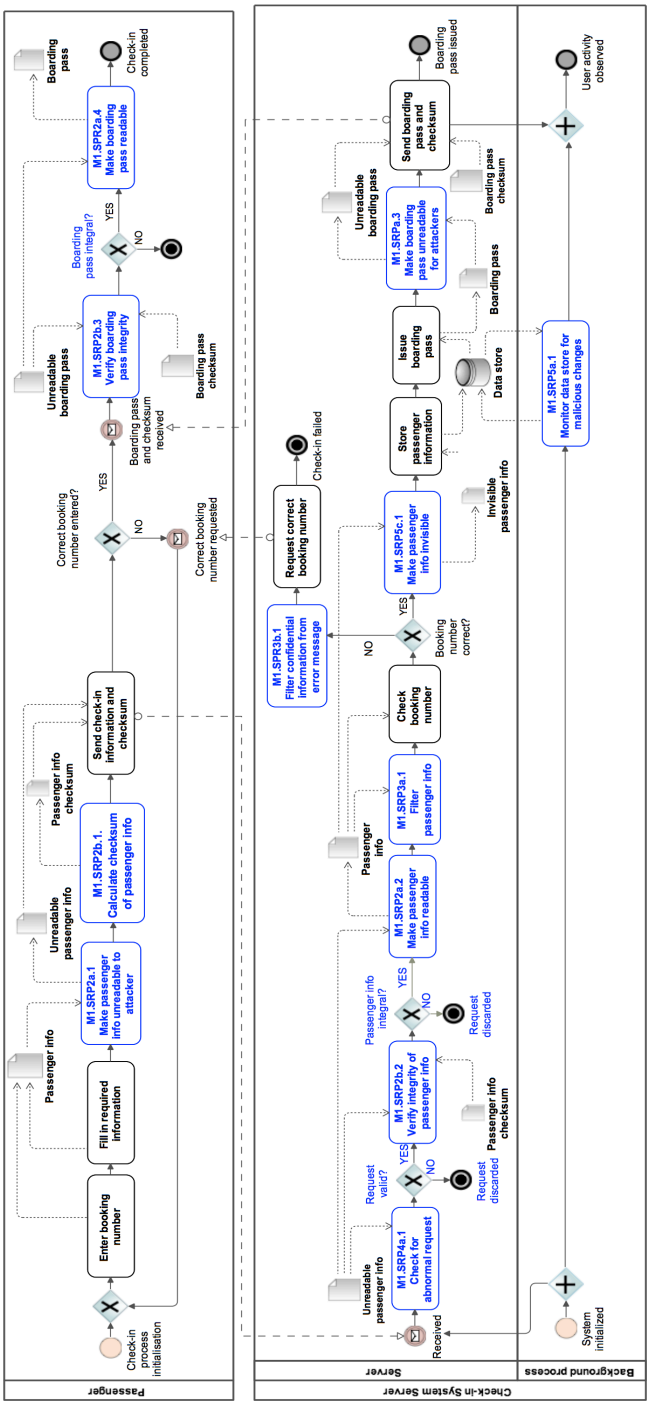

Fig. 8. Security requirements for *Passenger check-in process* derived using the security risk-oriented patterns.