



University of Tartu
Department of Computer Science

Naiad Hossain Khan

A PATTERN-BASED DEVELOPMENT OF SECURE BUSINESS PROCESSES

Master's Thesis

Supervisors: Dr. Raimundas Matulevičius, University of Tartu
Naved Ahmed, University of Tartu

Author:, June 2012

Supervisor:, June 2012

Supervisor:, June 2012

Approval for defence

Professor:, June 2012

Tartu, 2012

Abstract

Every security concerned enterprise selects its own security measures in order to avoid unexpected events and accidents. The main objective of these security measures is to protect the enterprise's own resources and assets from damage. Most of the time, the accidents or disasters take place in enterprise are similar in nature, and are caused by similar kind of vulnerabilities. However, many security analysts find it difficult to select the right security measure for a particular problem because the previous proven solutions are not properly documented. In this context Security Patterns could be helpful since they present the proven solutions that potentially could be reused in the similar situations.

In this thesis, we develop a set of ten Security Risk-oriented Patterns (SRP) and define the way how they could be used to define security countermeasures within the business process models. In principle, patterns are modelling language-independent. Moreover, to ease their application, we represent them in a graphical form using the Business Process Modelling Notation (BPMN) modelling approach.

We demonstrate the usability of the Security Risk-oriented Patterns (SRP) by applying them on two industrial business models. We present the quantitative analysis of their application. We show that Security Risk-oriented Patterns (SRP) help to determine security risks in business models and suggest rationale for security solutions.

The results of this research could potentially encourage the security analysts to follow pattern-based approach to develop secure business processes, thus, contributing to secure Information Systems (IS).

Thesis Outline

A PATTERN-BASED DEVELOPMENT OF SECURE BUSINESS PROCESSES	1
Abstract	3
List of Figures	8
List of Tables	10
Abbreviations and Acronyms	11
Chapter 1 Introduction	13
1.1 Motivation	13
1.2 Scope	13
1.3 Research Questions	13
1.4 Research Method	14
1.5 Thesis Structure	16
Chapter 2 Security Risk Management.....	19
2.1 Security Risk Management Approaches.....	19
2.2 Why ISSRM?.....	20
2.3 Information System Security Risk Management (ISSRM)	20
2.3.1 ISSRM Domain Model.....	20
2.3.2 ISSRM Process	21
2.4 Summary	22
Chapter 3 Business Process Management	23
3.1 Business Process Modelling Approaches	23
3.2 Business Process Modelling Notation (BPMN)	24
3.3 Why BPMN is needed for Security Analysis?	25
3.4 Risk Analysis: Three-level Approach.....	25
3.4.1 First Level: Process Modelling.....	26
3.4.2 Second Level: Risk analysis	27
3.4.3 Third Level: Security Requirement Elicitation.....	27
3.5 Alignment of BPMN with ISSRM	28
3.6 Aligning Business Process Modelling and Security Requirements.....	30
3.7 Summary	32
Chapter 4 Security Patterns	33
4.1 What is a Pattern?.....	33

4.2 Advantages of Pattern-based Security	33
4.3 Pattern Domains	34
4.4 Pattern Documentation	34
4.5 Security Risk-oriented Pattern Template.....	35
4.6 Summary	36
Chapter 5 Security Risk Classification.....	37
5.1 Vulnerability Taxonomies	37
5.2 Seven Pernicious Kingdoms of Vulnerability	38
5.3 Summary	40
Chapter 6 Security Risk-Oriented Patterns	43
6.1 SRP1: Securing data that flow between the business entities.....	43
6.2 SRP2: Securing input interface for allowing valid data enter into the business process.....	46
6.3 SRP3: Protecting the integrity of business activity by securing receiving interface	48
6.4 SRP4: Protecting IS from Denial Of Service (DOS) attack	51
6.5 SRP5: Applying multilevel access rights to retrieval interface	53
6.6 SRP6: Securing data confidentiality from unauthorised person in a data store.....	55
6.7 SRP7: Securing business activity from deadlock condition	57
6.8 SRP8: Ensuring atomicity of business transactions to protect data integrity	59
6.9 SRP9: Protecting data integrity in Time Of Check Time Of Use (TOCTOU) situation	61
6.10 SRP10: Preventing System Information Leakage	64
6.11 Summary	66
Chapter 7 Pattern Application.....	67
7.1 Step One: Occurrence identification.....	67
7.2 Step Two: Security criterion annotation	68
7.3 Step Three: Security risk requirement annotation	68
7.4 Step Four: Security requirement rationalisation.....	69
7.5 Summary	69
Chapter 8 Validation	73
8.1 Experiment Questions	73
8.2 Validation Methodology.....	73
8.3 Case Study I.....	74
8.3.1 Case Study Introduction	74

8.3.2 Process Quantification	74
8.3.3 SRP Application	74
8.3.4 Answers to Experiment Questions	75
8.4 Case Study II	75
8.4.1 Case Study Introduction	75
8.4.2 Process Quantification	75
8.4.3 SRP Application	76
8.4.4 Answers to Experiment Questions	76
8.5 Threats to Validity	76
8.6 Result Comparison	77
8.7 Summary	77
Chapter 9 Conclusion.....	81
9.1 Answer to Research Questions.....	81
9.2 Limitations.....	84
9.3 Future Work	84
Abstract eesti	85
Bibliography	86
Appendix	93
Occurrence of SRP1	93
Occurrence of SRP2	93
Occurrence of SRP3	94
Occurrence of SRP4	95
Occurrence of SRP5	95
Occurrence of SRP6	96
Occurrence of SRP7	97
Occurrence of SRP8	97
Occurrence of SRP9	98
Occurrence of SRP10	99

List of Figures

Figure 1 Research Method	15
Figure 2 Thesis structure	16
Figure 3 ISSRM Domain Model adapted from (Dubois, et al., 2010)	21
Figure 4 ISSRM Process, adapted from (Mayer, 2009)	22
Figure 5 Different Process Modelling Languages	24
Figure 6 Elements of BPMN (White, 2006).....	25
Figure 7 First Level : Process modelling.....	26
Figure 8 Second Level : Risk analysis	27
Figure 9 Third Level : Security requirement elicitation.....	28
Figure 10 Alignment of business process modelling and security requirements	31
Figure 11 Vulnerability Classification by (Landwehr, et al., 1994).....	38
Figure 12 Seven Pernicious Kingdoms of Vulnerability (Tsipenyuk, et al., 2005).....	39
Figure 13 Example business process.....	44
Figure 14 Potential threat analysis	45
Figure 15 Annotated security requirement.....	45
Figure 16 Example business process.....	47
Figure 17 Potential threat analysis	47
Figure 18 Annotated security requirement.....	48
Figure 19 Example business process.....	49
Figure 20 Potential threat analysis	50
Figure 21 Annotated security requirement.....	50
Figure 22 Example business process.....	52
Figure 23 Potential threat analysis	52
Figure 24 Annotated security requirement.....	52
Figure 25 Example business process.....	54
Figure 26 Potential threat analysis	54
Figure 27 Annotated security requirement.....	54
Figure 28 Example business process.....	56
Figure 29 Potential threat analysis	56
Figure 30 Annotated security requirement.....	56
Figure 31 Example business process.....	58
Figure 32 Potential threat analysis	58
Figure 33 Annotated security requirement.....	59
Figure 34 Example business process.....	60
Figure 35 Potential threat analysis	61
Figure 36 Annotated security requirement.....	61
Figure 37 Example business process.....	63
Figure 38 Potential threat analysis	63
Figure 39 Annotated security requirement.....	64
Figure 40 Example business process.....	65

Figure 41 Potential threat analysis 66

Figure 42 Annotated security requirement 66

Figure 43 Steps of SRP application guideline 67

Figure 44 Step 1 Occurrence Identification..... 67

Figure 45 Step 2 Security criterion annotation..... 68

Figure 46 Step 3 Security risk requirement annotation 68

Figure 47 Step 4 Security requirement rationalisation 69

Figure 48 Validation Methodology 73

Figure 49 Occurrence of SRP1 93

Figure 50 Occurrence of SRP2..... 94

Figure 51 Occurrence of SRP3..... 94

Figure 52 Occurrence of SRP4..... 95

Figure 53 Occurrence of SRP5..... 96

Figure 54 Occurrence of SRP6..... 96

Figure 55 Occurrence of SRP7..... 97

Figure 56 Occurrence of SRP8..... 98

Figure 57 Occurrence of SRP9..... 99

Figure 58 Occurrence of SRP10..... 100

List of Tables

Table 1 BPMN & ISSRM Alignment adapted from (Altuhhova, et al., 2012) 29

Table 2 Security Risk-oriented Pattern Template adapted from (Ahmed & Matulevičius, 2011) 35

Table 3 Quantitative description of Case Study I business process model 74

Table 4 SRP occurrences in business process of Case Study I 74

Table 5 Quantitative description of Case Study II business process model 75

Table 6 SRP occurrences in business process of Case Study II 76

Abbreviations and Acronyms

API	Application Programming Interface
AURUM	Automated Risk and Utility Management
BPM	Business Process Management
BPMN	Business Process Modelling Notation
DOS	Denial Of Service
IS	Information Systems
ISSRM	Information System Security Risk Management
MLS	Multi-Level Security
SQL	Structured Query Language
SRP	Security Risk-oriented Pattern
TOCTOU	Time Of Check Time Of Use
XML	Extensible Markup Language
XPath	XML Path Language
XSS	Cross-site scripting

Chapter 1 Introduction

Business processes are vulnerable to various security risks. Most of the time, the business analysts remain busy in optimising the processes, rather than focusing on security aspects. Security is often given less priority, and it is addressed at later stages (e.g. implementation stage) of process development. We wish to move security to an early stage of business process development and propose a pattern-based solution for accomplishing it.

1.1 Motivation

Today's business enterprises depend on Information Technology (IT). The business decision-maker would like all technical activities - which affect their enterprise, to be consistent with accomplishing the enterprise's goals. However, reality often frustrates them by challenging with unexpected and unfortunate events. Industrial information leakage, bank credit card frauds are common news in daily newspapers (Markoff, 2012) (Gaudin, 2007). All these incidents harm the reputation of business companies and as a result, they suffer financial loss. This scenario inspires us to conduct research for finding pattern-based security solution - which could be helpful for saving companies and industries business processes from future tragedies.

1.2 Scope

The research utilises Information System Security Risk Management (ISSRM) (Dubois, et al., 2010) - a risk analysis framework, to analyse business processes for security issues. It also uses Business Process Modelling Notation (BPMN) (OMG, 2012) to understand the performance collaborations and business transactions present in business processes. The research discusses about the origin of security patterns, their domains and documentation method. Two major alignments: between ISSRM and BPMN, and between ISSRM and Security Pattern, are used for developing Security Risk-oriented Patterns (SRP). Different vulnerability taxonomies (Tsipenyuk, et al., 2005), (Landwehr, et al., 1994) and (R.Abbott, et al., 1975) are discussed to estimate the scope of SRP development. Ten Security Risk-oriented Patterns (SRP) are developed for the purpose of addressing business process security risks. The research also proposes SRP application guideline for the development of secure business processes.

1.3 Research Questions

We present three research questions which help to achieve our research objective. The questions should also help the readers to understand the overall purpose and contribution of this research-based thesis. Here, we explain the questions and their answers are given at the end of this report.

RQ 1: *How to make business process secure?*

This question stands as the main research problem of this thesis. Previously, we have mentioned different types of industrial frauds and IT related security issues - which often act as headache of business owners. As an answer to this question, how business analyst and security analyst could use Security Risk-oriented Patterns (SRP) to develop secured business processes will be described.

RQ 2: *What are the Security Risk-oriented Patterns to secure business processes?*

The research focuses on developing Security Risk-oriented Patterns (SRP). These SRPs are used to address security risks - which are present in business processes. The answer of this question will provide the descriptions of ten Security Risk-oriented Patterns (SRP).

RQ 3: *How do the Security Risk-oriented Patterns help to secure business processes?*

This question seeks the answer regarding the usefulness of Security Risk-oriented Pattern (SRP) in business process security risk mitigation. The answer will present the SRP application guideline and the results after applying them in two case studies.

1.4 Research Method

In figure 1, we show our research method which contains three stages (orange coloured boxes), five primary inputs (blue coloured boxes) and four outcomes (pink coloured boxes). Three of these outcomes are also considered as derived inputs for latter two stages (Knowledge, Security Risk-oriented Patterns, and Security Risk-oriented Pattern Application Guideline). The research stages are described below:

Stage One: Background Study. Various research papers, books are read and analysed during the research preparation period. The background materials are divided into two types: Security risk mitigation related and Business process management (BPM) related literatures. Security risk mitigation related literatures help to understand different types of vulnerabilities, threats, security risks and negative impacts. These also introduce with Security risk management frameworks and Pattern-based security solutions. On the other hand, Business process management related literatures describe various business process modelling approaches, the relationships between the process execution and the used resources, and many other aspects of Business process management (BPM). The combined knowledge - acquired from both of these literature types, assists us to carry on the research.

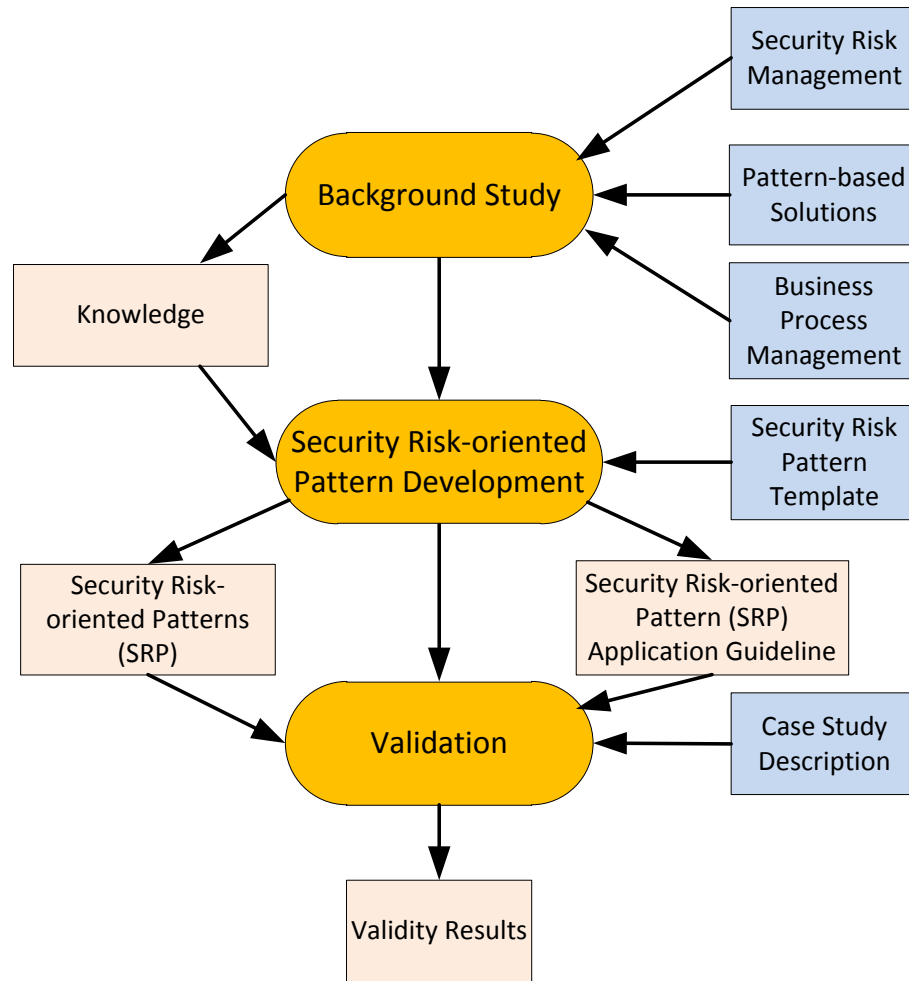


Figure 1 Research Method

Stage Two: Security Risk-oriented Pattern Development. In this stage, with the help of Security Risk-oriented Pattern (SRP) template (Ahmed & Matulevičius, 2011), ten Security Risk-oriented Patterns (SRP) are prepared for addressing eight categories of vulnerabilities (Tsipenyuk, et al., 2005). The Security Risk-oriented Pattern (SRP) application guideline is also proposed.

Stage Three: Validation. In the final stage, The SRPs are applied according to the proposed SRP application guideline in two case studies. The quantitative results of the validation are presented in tabular format.

1.5 Thesis Structure

Figure 2 presents the structure of the thesis. Each box represents a single chapter, and the boxes marked with the same colour belong to the same part of the thesis report.

The report is consists of nine chapters. Chapter 1 presents the overview of thesis scope, motivation, research questions and research method. Part I consists of four chapters. Chapter 2 discusses about Security Risk Management. This discussion includes different risk analysis frameworks, especially, Information System Security Risk Management (ISSRM) (Dubois, et al., 2010), its domain model and ISSRM process. In chapter 3, ISSRM framework is aligned with Business Process Modelling Notation (BPMN) (Altuhhova, et al., 2012). Chapter 4 describes security patterns and the benefits of using them. It also presents a complete Security Risk-oriented Pattern Template (Ahmed & Matulevičius, 2011) aligned with ISSRM domain elements. Next, chapter 5 introduces with different categories of vulnerabilities responsible for causing different types of risks. Part II includes two chapters. Chapter 6 presents ten Security Risk-oriented Patterns (SRP). Their application guideline is describes in chapter 7. In chapter 8, ten SRPs are validated in two case studies. Finally, chapter 9 provides the conclusion and future research directions.

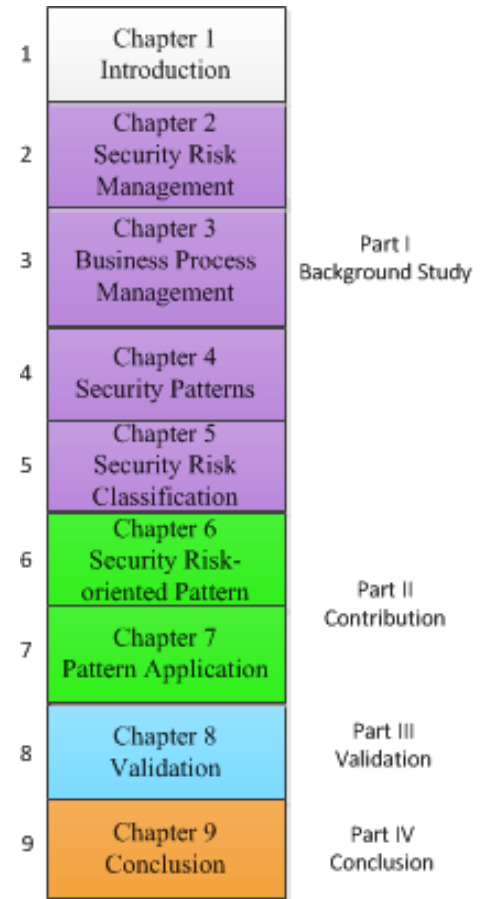


Figure 2 Thesis structure

Part I

Background Study

This part surveys the state-of-the-art of security risk management frameworks, business process modelling approaches, security patterns and the vulnerability taxonomies. It consists of four chapters.

The second chapter discusses different security risk management approaches and frameworks. Advantages of Information System Security Risk management (ISSRM) (Dubois, et al., 2010) approach over CORAS (Lund, et al., 2011) and AURUM (Ekelhart, et al., 2009) are discussed. ISSRM domain model and ISSRM process are explained further in this chapter.

The third chapter is about Business Process Management. It focuses on different business process modelling approaches which are currently being used by process analysts. Different criteria of good modelling languages are mentioned. The chapter shows the potentiality of Business Process Modelling Notation (BPMN) (OMG, 2012) in performing business process security risk analysis by presenting the Three-level risk analysis approach. ISSRM & BPMN (Altuhhova, et al., 2012) alignment explains how the current limitations of BPMN in security risk analysis could be resolved. At the end of the chapter, an alignment between business process modelling and security requirements is also proposed.

The fourth chapter begins with a short history of pattern-based solution in software engineering. This is followed by the benefits of using pattern-based security approach. The chapter discusses about different pattern domains and structure of security pattern. We are introduced with Security Risk-oriented Pattern Template proposed by (Ahmed & Matulevičius, 2011). This template is used to develop ten Security Risk-oriented Patterns (SRP).

The fifth chapter surveys different taxonomies of vulnerabilities. The taxonomy model proposed in '*Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors*' (Tsipenyuk, et al., 2005) is chosen to estimate the scope of SRP coverage area. Eight vulnerability categories are discussed in details which are responsible for causing different categories of security risks, and these risks are addressed by the SRPs.

Chapter 2 Security Risk Management

There exist several approaches for security risk management. In this chapter, we review three of these: AURUM (Ekelhart, et al., 2009), CORAS (Lund, et al., 2011) and ISSRM (Dubois, et al., 2010) . We shed light on all these frameworks and present our arguments behind choosing Information System Security Risk Management (ISSRM) as the framework for security risk management in our research. In addition, ISSRM is described in detail which should help to understand how to determine valuable assets, potential security risks and countermeasures to these risks.

2.1 Security Risk Management Approaches

Today's researchers view security risk management mainly in two different ways. The first group of researchers takes the asset-based evaluation approach (Dubois, et al., 2010). In this approach, the researchers estimate risk on the basis of asset analysis. By doing this, business process analysts are able to trace the risk origins - which are the vulnerable assets. On the other hand, the second group of researchers focuses on the value of the business processes, rather than on the assets (Khanmohammadi, 2010). They believe, since enterprises' earned revenue is directly related to their core business processes, so a business process renders more value to an enterprise than its assets involved in accomplishing the process, and therefore, they should focus on business process-related risk analysis. In fact, this second approach evaluates assets too, but rather than direct assessment, it evaluates indirectly.

Other approaches like CORAS (Lund, et al., 2011) uses customised language for risk modelling and it describes how the language could be used effectively for capturing relevant information during security analysis. CORAS presents the result of analysis in UML and performs the process of analysis in eight different steps (CORAS, 2012).

Automated Risk and Utility Management or AURUM - proposed by (Ekelhart, et al., 2009), supports NISTSP 800-30 (Jakoubi, et al., 2009) risk management standards. However, NISTSP 800-30 divides the risk management process into three major steps (risk assessment, risk evaluation and risk mitigation); but in AURUM, ten distinct steps are being followed. Besides, AURUM obeys the security ontology described in (Ekelhart, et al., 2007).

Information System Security Risk Management (ISSRM) (Dubois, et al., 2010) uses security modelling language for counting security issues and correlates risk management task throughout all stages of Information System development. Risk management process is carried out from three different conceptual levels such as: Asset-related, Risk-related and Risk treatment-related concepts. Risk management process could continue while developing Information System, which is a unique feature of this risk management approach.

In these above mentioned approaches, two common steps are performed during security risk management. The first step is to identify and estimate the risks present in a business process on the basis of empirical data, knowledge and expertise. This step also includes prioritising the risks according to their severities. In the second step, security analysts search for security solutions, compare them according to their cost

and feasibility to mitigate the identified risks. These two steps could only be performed effectively when the sufficient empirical data and expertise are present.

2.2 Why ISSRM?

Considering various pros and cons of different security risk management approaches described in the previous section, we come up with the following reasons for choosing ISSRM as the framework for security risk management in this research:

- The domain model of ISSRM clearly expresses the relationships between all the considered entities present in business process during risk assessment (see ISSRM domain model in figure 3).
- ISSRM model is compliant with the current security standards e.g. (Karagiannis, et al., 2007)
- ISSRM comprises the complete risk management. In other words, it not only identifies and defines the risk, but also shows the risk mitigation techniques.
- Current risk management approaches require existing Information Systems (IS) in order to perform risk analysis. In contrast, ISSRM is also applicable for developing future secured business process using patterns templates (Ahmed, et al., 2012).

2.3 Information System Security Risk Management (ISSRM)

2.3.1 ISSRM Domain Model

Information System Security Risk management (ISSRM) (Dubois, et al., 2010) is methodological tool which assists organisations in making decisions related to the security of their Information Systems. ISSRM not only helps the security pattern development process, but also ensures their optimum usability.

The domain model shown in Figure 3 delineates the main ISSRM concepts, their relationships and corresponding definitions. The model describes three principle groups of concepts: *asset-related concepts*, *risk-related concepts* and *risk treatment-related concepts* - marked with yellow, orange and green colours respectively.

The first group, *asset-related concepts* describes the important assets - which need to be protected. The asset is comprised of two main types: the *business asset* – which is defined as information, process, and skill necessary for achieving organisation’s objectives; and *IS asset* – which has value to the organisation and supports business asset. *Security criterion*, such as *confidentiality*, *integrity* and *availability* characterise the security needs of business asset.

The second group is *risk-related concepts*. *Risk* is a combination of *event* and *impact*. An *event* is another combination of a *threat* and one or more *vulnerabilities* - which leads to *impact*. Here the *impact* means an undesirable consequence of a risk which harms *assets* of an organisation when a *threat* is successfully accomplished. *Threat* exploits the weaknesses of the *IS asset* which are referred as *vulnerabilities*. A *threat agent* is someone with the ability to cause intentional harm to *IS assets*. Moreover, a *threat agent* uses *attack method* - a standard mean by which he executes *threat*.

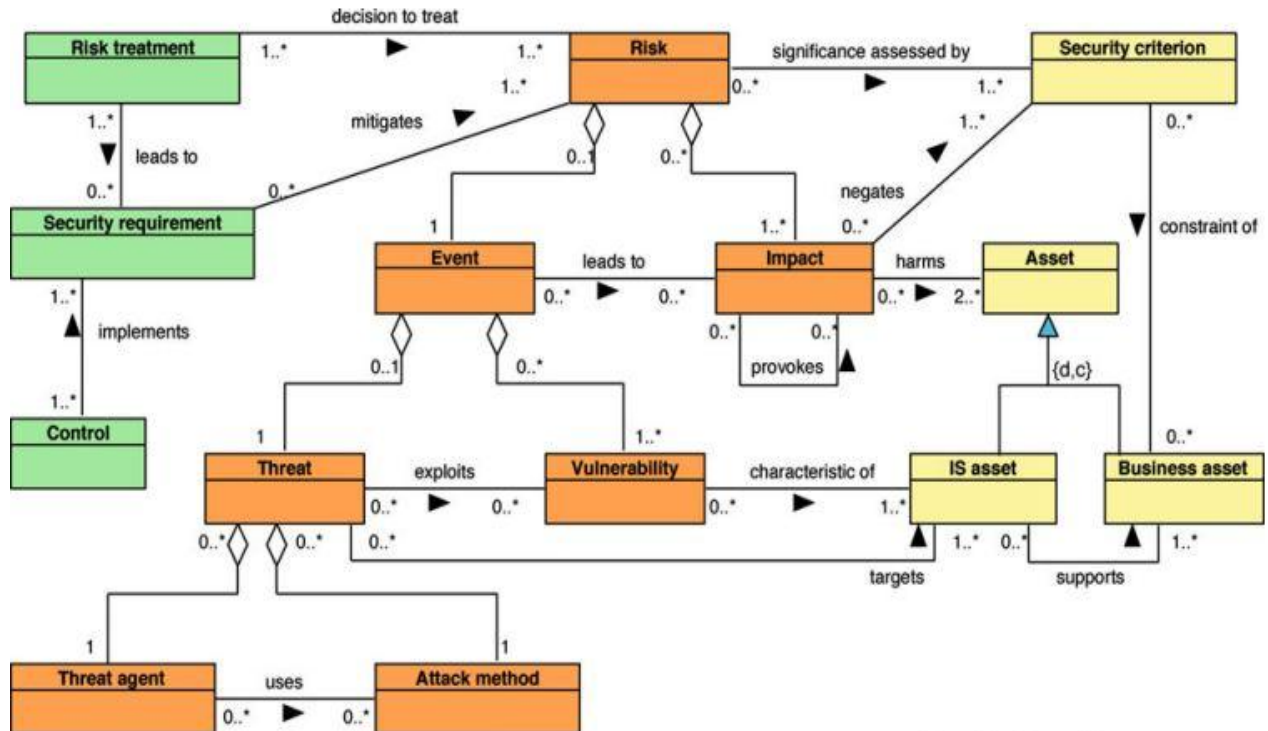


Figure 3 ISSRM Domain Model adapted from (Dubois, et al., 2010)

The third group is *risk treatment- related concepts*. *Risk treatments* are the decisions to treat the identified *risks*. Generally, each risk treatment fulfils a security need. The categories of risk treatment decisions include: *risk avoidance* - decision not to become available in a risk; *risk reduction* - action to reduce the probability of negative consequence; *risk transfer* - decision to share the burden of loss from a risk to another party; *risk retention* - accepting the burden of loss from a risk. The *risk treatment* is implemented by a designed mean to improve security, specified by a *security requirement*. A *control* represents the mean to improve the security by implementing *security requirements*.

2.3.2 ISSRM Process

ISSRM process (shown in Figure 4) describes the activities needed to identify, monitor and control security risk. Within the process, a *risk* is defined as any future events, which may prevent one from meeting the enterprises goals. This process helps to identify risk, quantify the impact and take actions to prevent it from occurring in the business process.

ISSRM process is composed of six steps. The first step is dedicated to *context and asset identification*. It starts with the analysis of the organisation, its environment and assets - which need protection. It proceeds to the *determination of security objectives* (e.g. integrity, confidentiality or availability). Third step is *risk analysis and assessment*. This step's purpose is to identify and estimate risks qualitative or quantitatively. Following is the *risk treatment*, which is an activity for selecting and implementing measures to modify

the risk. Risk treatment includes risk control/mitigation, but also extends further to, for example, risk avoidance, risk transfer, risk retention etc. After this, the next step is performed to *define security requirements*, i.e. the security solutions to mitigate the risks. Finally, it is necessary to select and implement the *countermeasures/controls* within the enterprises business processes.

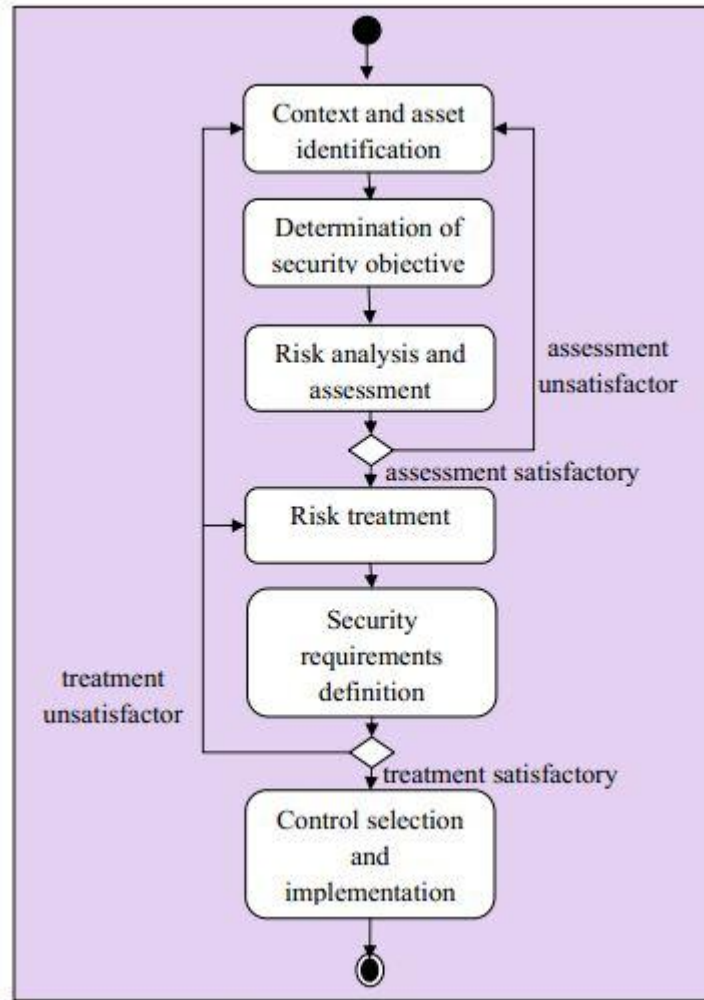


Figure 4 ISSRM Process, adapted from (Mayer, 2009)

2.4 Summary

In this chapter, we survey few security risk management approaches. We select the ISSRM domain model. It defines interdependencies between various security domain elements. The ISSRM process describes the way how the domain model could be applied. The knowledge gained from this chapter contributes to the research when enterprise business processes are analysed for security issues.

Chapter 3 Business Process Management

Business process is a procedure which is found in business organisation. From purchasing a coffee from Starbucks to importing industrial goods from foreign country - all these activities go through a route of chained tasks, which is considered as business process. For maximising the profit in highly competitive business market, companies often focus on exploring and managing their own business processes – which is referred as Business Process Management (BPM). Business process analysts use different modelling languages to model the processes. In this chapter, we discuss about different modelling approaches which are currently being followed in business process management. We choose Business Process Modelling Notation (BPMN) (OMG, 2012) for analysing business processes for our research. In addition, we also show how BPMN and ISSRM could be used for performing business security risk analysis.

3.1 Business Process Modelling Approaches

Business processes present in modern companies are often complex and complicated than the past. So business analysts often use to model the processes by using different modelling languages to understand the processes clearly. Following are some benefits of business process modelling (Ko, 2009):

- Business process modelling language enhances the perceptibility and knowledge on company's inner activities.
- Business analyst's quest for identifying bottlenecks in business process becomes an easy task by process modelling.
- It helps to detect and identify potential areas - which need optimization.
- Reduces delay or lead-times.
- Business process modelling language describes who is responsible for what duty in a company.
- Finally, it is an effective tool for fraud prevention and auditing of regulations compliance.

Paradoxically, J.A. Zachman in his article 'A Framework for Information System Architecture' (Zachman, 1987) contends that it is impossible for a single model to capture all the important features of a business enterprise. Each of the different modelling approaches has its own benefits and drawbacks. Stefan Haberl mentions a group of seven criteria for the evaluation of process modelling methodologies (Dufresne & Martin, 2003):

- It should be capable of modelling all the complexities of business processes which include: sequencing, branching, looping, concurrency constructs (fork and synchronize), timeouts, exception handling etc.
- It should have a method of distinguishing roles and assigning them the different duties.
- A clear-cut graphical representation of the language should exist.
- It should be able to show how a process could be undone.
- It should describe how process instances can be started and followed throughout their execution.
- It must possess the characteristics of good demonstration of the business process. It should be capable of raising the interests of external users, especially the interests of the business process analysts.
- Lastly, the language should not mingle in details of communications protocols.

Some of the process modelling languages are:

- Flow charts (IBM, 1969)
- Data Flow Diagrams (Draw, 2012)
- Control Flow Diagrams
- Unified Modelling Language (UML) (OMG, 2006)
- Business Process Modelling Notation (BPMN) (OMG, 2012)

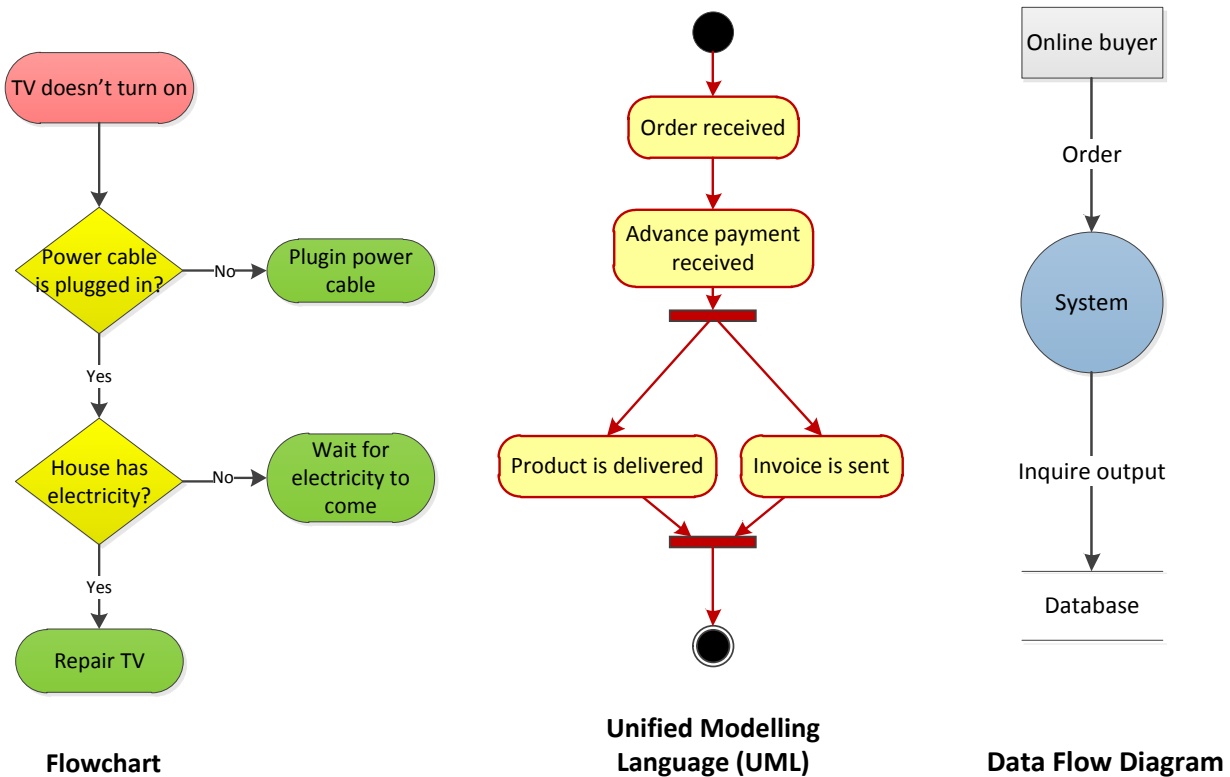


Figure 5 Different Process Modelling Languages

Although the former languages are still quite popular and widely being used, the latter two languages are gaining popularity day by day because of their compliance with previously mentioned criteria.

3.2 Business Process Modelling Notation (BPMN)

Business Process Modelling Notation (BPMN) is a language for developing and describing business process models. There are three main different levels (White, 2006) of process modelling. They are: *Process Maps* - which are flow charts of the business activities; *Process Descriptions* - additional information added on top of flow chart, but not sufficient for fully defining actual performance; and *Process Models* - additional information which can be used to simulate or execute the whole business process. BPMN addresses each of these levels, which is one of the big reasons behind its superiority over other process modelling languages. BPMN is developed by Business Process Management Initiative

(BPMI) (Group, 2008) and later merged with Object Management Group (OMG) (Group, 2012) - which now maintains it. The first specification BPMN 1.0 was released in May, 2004 (White, 2006). The latest version BPMN 2.0 was released to public in January 2011.

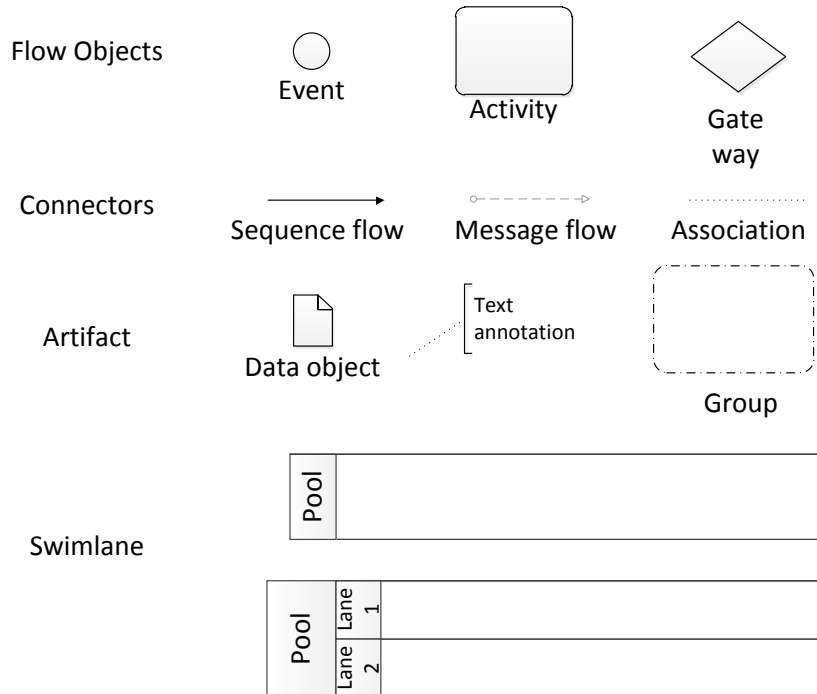


Figure 6 Elements of BPMN (White, 2006)

3.3 Why BPMN is needed for Security Analysis?

Security risk analysis in business processes is important for information system development. The enterprises are vulnerable to potential security risks due to the nature of inherent risks in their routine operations and interactions with stakeholders (Ahmed & Matulevičius, 2011). Detecting and perceiving the subtle relationships between the risks and vulnerabilities are essential for preparing risk mitigation solutions. In order to perform thorough and complete analysis, Business Process Modelling Notation (BPMN) is widely used for the understanding of internal business operations and their corresponding communications.

3.4 Risk Analysis: Three-level Approach

In this section, we show how a simple business process could be analysed with the help of BPMN and ISSRM for security risks by a three-level risk analysis method (Figure 7, 8, 9).

3.4.1 First Level: Process Modelling

In the first level (Figure 7), a complete business process model is developed by using BPMN on the basis of a business process description. Most of the time, the entire business process is complex and lengthy. Therefore, a short part of a total business process is considered in this example for the ease of understanding.

The following paragraph describes an example business process:

A company holds a couple of branches or subsidiaries around the world. The offshore branches need to keep frequent contact (e.g. exchanging documents, phone conversations between employees and so on) with their head office in order to perform their own business activities. In this modern digital era, instead of exchanging the documents in hard format, the head office and the branches prefer to exchange documents between them in digital format. Moreover, decision makers – who are sitting in the head office, take important decisions on the basis of the document's data for selecting future business strategies.

In first step, we draw this business process scenario using BPMN (Figure 7). Both of the employees working in Head office and one of the Branch offices are shown by using two separate pools. This part of the business process starts when an employee – who works in the off-shore branch office (upper pool), sends a document (e.g. .doc, .docx file) to the head office - which is located in another country (lower pool). After it has been successfully received by the head office's server, the employees (i.e. decision makers) retrieve the document using their computers and prepare it for later discussion in which they take business decision. After this, the process reaches to an end. It is essential to identify which one of these depicted activities is the most important. Among the three activities depicted here, the *Make business decision* activity could be considered having the top most importance, since on the basis of the outcome of business decision, the company operates in future. So, the company should ensure the *integrity* and perfection of this activity.

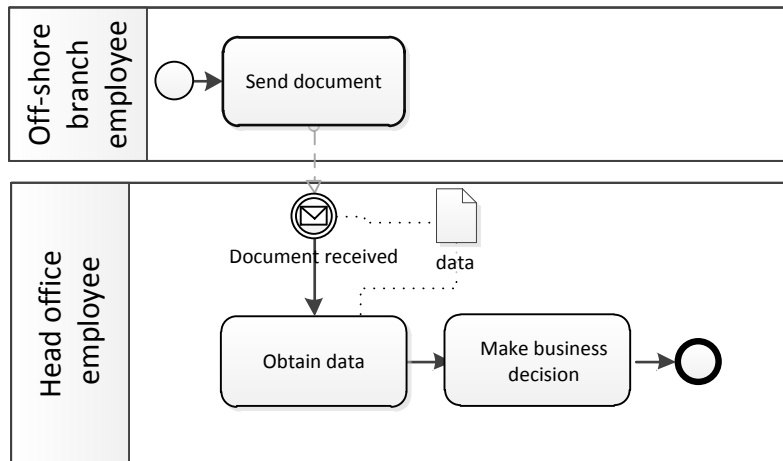


Figure 7 First Level : Process modelling

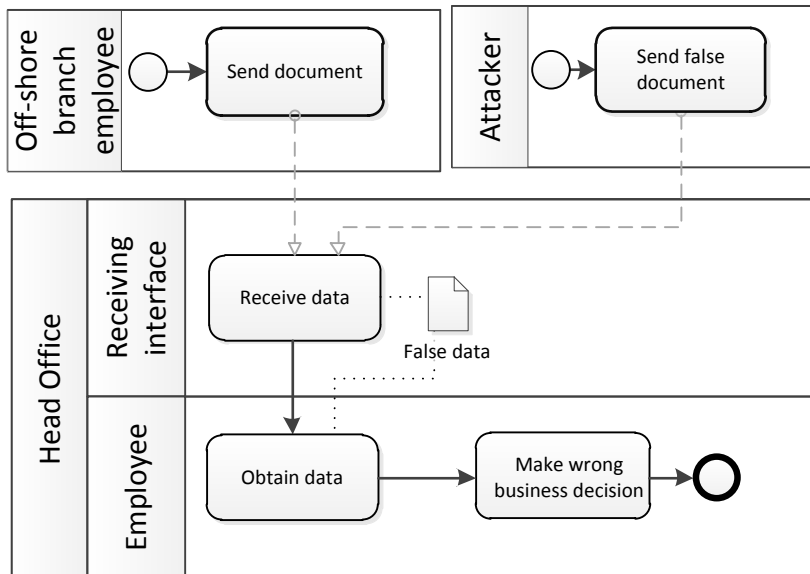


Figure 8 Second Level : Risk analysis

3.4.2 Second Level: Risk analysis

In this level (Figure 8), the main task is to analyse the business process scenario developed in the previous level for possible vulnerability and security breaches. How the *integrity* of the *Make business decision* activity could be breached? Generally, the head office employees assume that the document has arrived from their branch office by observing the email address, but they are unable to confirm that whether this was actually sent by the off-shore branch office employee - who is responsible for sending the document. It could happen that the off-shore branch office employee's computer has been hacked by a hacker - who can send falsified data in order to cause harm to the company (shown in figure 9). Or, while transferring the document, the transmission medium could be intercepted by an adversary - who can modify the actual data of the document and send it to the receiving head office. After the risk has been identified (in this example, we consider the former risk), the next task is to find the exact IS asset in order to track the origin of the vulnerability present in the business process. In fact, the falsified document is first received by the server or the receiving interface of head office. So another separate lane: Receiving interface is added to the Head office pool (Figure 8).

3.4.3 Third Level: Security Requirement Elicitation

In the third level (Figure 9), possible prevention method or technique is pursued in order to mitigate the vulnerability identified or discovered in the second level. After searching through security risk management literatures and related scenarios, it appears that *digital signature scheme* can help to mitigate this security risk. Under this scheme, before sending the document, the branch office employee needs to sign it digitally. Later, when the document is received by the head office's receiving interface, the signature is verified. This ensures the identity and integrity of document. If the verification result is

positive, the business process flow proceeds into the next activity. Otherwise, it becomes cancelled and reaches to an end. The security requirement ‘Verify the identity of digital document sender’ has been added in figure 9.

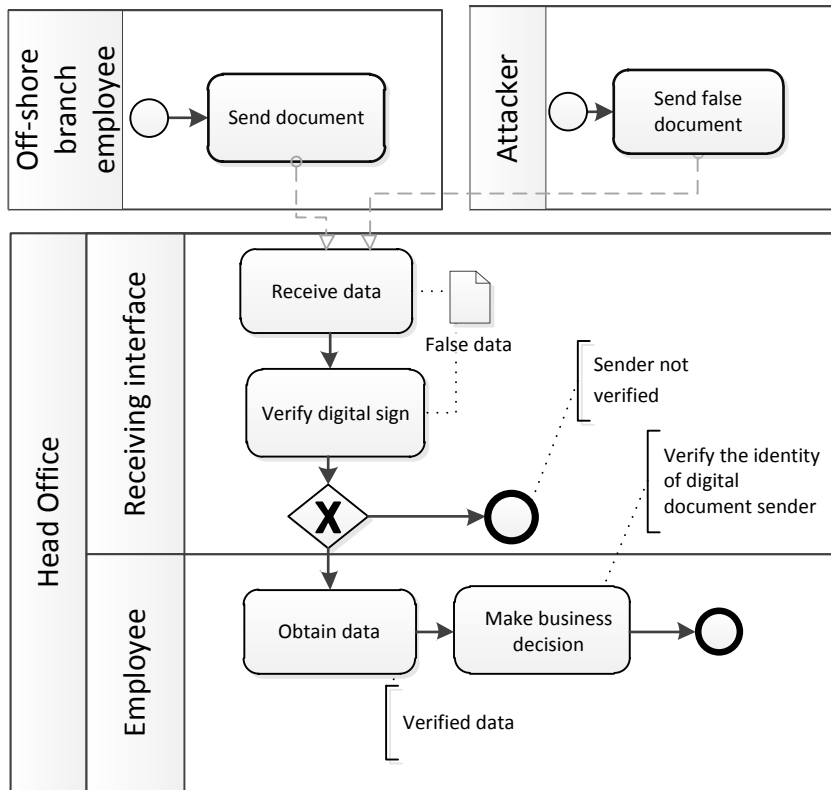


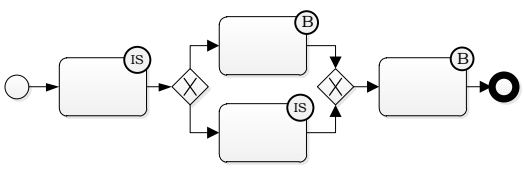




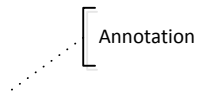


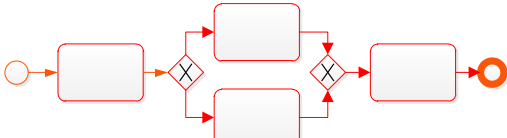

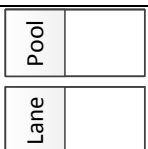


Figure 9 Third Level : Security requirement elicitation

3.5 Alignment of BPMN with ISSRM

In the previous analysis, BPMN proved its potential for business security risk analysis. In ‘Towards Definition of Secure Business Processes’ research paper, Altuhhova, Matulevičius and Ahmed aligned BPMN constructs with the elements of ISSRM (Altuhhova, et al., 2012) (Table 1). This relationship is a recent proposed idea, but this alignment helps risk analysts to scrutinise business processes using BPMN and the ISSRM framework.

Table 1 BPMN & ISSRM Alignment adapted from (Altuhhova, et al., 2012)

	The ISSRM domain model	BPMN constructs	Syntax
Asset-related concepts	Asset	Combination of Flow Objects (Event, Gateway, Tasks) using sequence flow For Business assets  For IS assets 	
	Business asset	Data object	
	IS asset	Data store Containers (Pool and Lanes)	 
	Security criteria (with Security objectives)	Lock sign consisting of three different values: <ul style="list-style-type: none"> • c - confidentiality • i - integrity • a - availability Locks can be associated with annotations	 
Risk-related concepts	Vulnerability	Vulnerability point and Association Flow that points to Annotation	 
	Attack method	Combination of Flow Objects (Event, Gateway, Task) using Sequence Flow	
	Impact	Unlock sign consisting of three different values: <ul style="list-style-type: none"> • c - Breach of confidentiality • i - Breach of integrity • a - Breach of availability 	
	Threat agent	Containers (Pool and Lanes)	

	Threat	Combination of constructs for Threat agent and Attack method	
	Event	Combination of construct for Threat and Vulnerability	
	Risk	Combination of Event and Impact	
Risk treatment-related concepts	Risk treatment	-	-
	Security requirement	Combination of Flow Objects using Sequence Flow (In this report, we simply use notations to define Security requirement)	
	Control	-	-

From the table, it is prominent that not all the elements of ISSRM could be aligned or mapped with BPMN constructs. Currently, BPMN 2.0 is being used by the business analysts. In future, additional constructs of BPMN can be proposed to fill up these gaps.

3.6 Aligning Business Process Modelling and Security Requirements

Generally, business analyst and security analyst collaborate with each other to address the security risks – which are present in the business process. By integrating security requirements when modelling the business process, we could address security at an early stage of business process development. In Figure 10, we present a method to align business process modelling with security requirements. This method has total 7 steps. The rectangle boxes symbolise the output and inputs of the different steps. Boxes are marked

with two different colours in order to display their associations with two different groups (i.e. business analysts and security analysts).

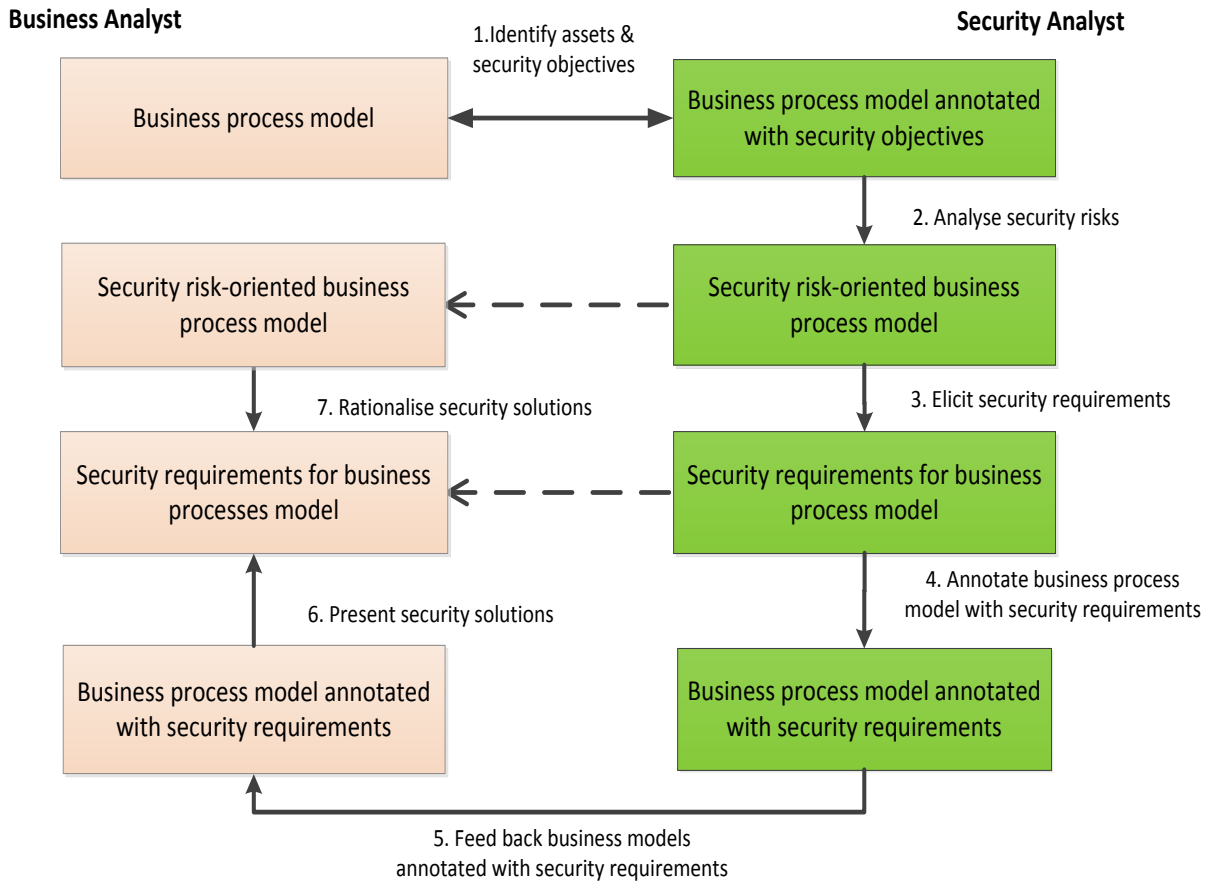


Figure 10 Alignment of business process modelling and security requirements

The steps of the aligning method are described below:

1. **Identify assets and security objectives.** In this step, a business analyst introduces the business process model to a security analyst. They mutually share their ideas, opinions and comments to identify business and IS assets in the process model. Besides, they also define the security objectives for the identified assets.
2. **Analyse security risks.** The security analyst is solely responsible for performing this step. He uses risk analysis framework (e.g. ISSRM) in order to identify the potential threats, vulnerabilities and risks to the assets. The first and second levels of the three-level risk analysis approach – which is presented in section 3.4, could serve as a potential example of such security risk analysis.
3. **Elicit security requirements.** During this step, security analyst seeks for the risk treatment decisions in order to mitigate (i.e. reduce, avoid, transfer etc.) the identified risks.
4. **Annotate business process model with security requirements.** Since the primary goal of business process models is to describe the process work-flow, therefore, the security analyst refrains from adding new modelling constructs (e.g. activity, gateway etc.) into the model to represent security

requirements. Instead, he uses annotations in the business process model to represent those requirements.

5. **Feed back business models annotated with security requirements.** In this step, security analyst gives back the business process model to the business analyst with security requirement annotated on it. At this point, the business analyst is able to realise the complete security need for his business process.
6. **Present security solutions.** After receiving the annotated model from security analyst, the business analyst starts the initiative for fulfilling the proposed security requirements. However, security requirements come with a trade-off between cost, time and process complexity. Furthermore, a single security requirement can be fulfilled by using any of the multiple available solutions. Therefore, business analyst considers prioritising the security requirements and examining all the available security solutions presented by the security analyst.
7. **Rationalise security solutions.** Sometimes, it can be hard to prioritise the security requirements. In other words, without clear understanding of the risk consequences (i.e. impact of risk), it is difficult to choose between multiple security requirements. During this step, security analyst helps business analyst prioritise security requirements by providing him detail explanations and rationales.

3.7 Summary

This chapter presents the knowledge and technique on how an ordinary business process can be modelled and analysed for security risks using BPMN. The benefit of using BPMN as the language for analysis is explained at the beginning of the chapter. The three-level security analysis approach could be considered as a security risk analysis methodology. In addition, the alignment between BPMN and ISSRM helps to develop business processes and understand security requirements in parallel. This alignment is used for the development of Security Risk-oriented Patterns (SRP) in chapter 6. At the end of this chapter, a method is proposed for aligning Business Process Modelling with Security Requirements. This method could help to start addressing security risks at an early stage of business process development.

Chapter 4 Security Patterns

In pattern-oriented software engineering, knowledge is collected from the relevant domain - which provides the basis for solving problems. The majority of software security issues often do not require new solutions. The developers reuse similar solutions - which are already successfully implemented by others to mitigate security risks. In this chapter, we discuss about patterns, the benefits of using pattern-based solution in software engineering and their domains. We also see how the security risks, vulnerabilities, solutions and controls can be presented in formal ways. This formal representation is referred as *Security Pattern*. The knowledge gained from the chapter helps to develop Security Risk-oriented Patterns (SRP) for our research.

4.1 What is a Pattern?

A pattern represents a proven solution of a problem - which arises within a certain context. In 1994, the work by the *Gang-of-Four* (Gamma, et al., 1994) proposed pattern as a 'new concept'. Software developers hoped that patterns would help them to solve difficult problems with well-defined solutions. However, the scope of those patterns had only a small impact on total software or system architecture. This limitation was overcome by the *POSA team* (Schmidt, et al., 2000). Since then, patterns appear into many other specific areas such as: concurrent and networked systems, human-computer interaction, resource management etc. Security is another area of major interest for patterns and following is a definition of security pattern by researcher Markus Schumacher:

"A security pattern describes a particular recurring security problem that arises in specific contexts, and presents a well-proven generic solution for it. The solution consists of a set of interacting roles that can be arranged into multiple concrete design structures, as well as a process to create one particular such structure." (Schumacher, 2003) (Schumacher, et al., 2004)

4.2 Advantages of Pattern-based Security

There are many benefits which promote the use of security patterns in different domains (Schumacher, et al., 2004):

- Security patterns describe basic security knowledge in a formal structured way.
- Significant time is not required for the software developers to understand security pattern representation.
- Using patterns to capture security knowledge helps to improve the integration of security into systems and enterprises, since patterns have been already being used to capture organisation and system engineering knowledge for quite a long period.
- Finally, pattern emphasises not only on the solution, but also on the cause of the problem.

4.3 Pattern Domains

Security patterns are applied in various areas of our life. Here, we discuss some notable pattern usage domains which are mentioned in (Schumacher, et al., 2004)

Enterprise Security and Risk Management. Each and every enterprise has some missions, and it focuses on addressing security issues - which are related to these missions. The pattern addresses enterprise-wide security issues. Asset evaluation, vulnerability assessment, risk determination are the main aspects analysed by the patterns.

Identification & Authentication (I&A). Pattern guides us to select the right solution when we have multiple available biometrics I&A alternatives. It helps the both security analysts to enforce constraints on system passwords, and the system users to select passwords for password authentication system. Face recognition, Iris recognition, Retinal scanning, Signature verification, PKI design variables, Speaker verification are some of the examples which can be successfully implemented using security patterns.

Operating System Access Control. Pattern discusses about the authentication - which is needed during file access. It shows how a subject can be approved or authorised to gain access to an object in a specific way, and how to verify that the requestor is not an imposter.

Firewall Architecture. Before selecting any firewall, it is required to consider the trades-offs between the speed, complexity and security of various types of firewalls such as: Packet filter firewall, Proxy-based firewall, Stateful firewall etc. The patterns describe all these different types of firewalls and guide people to select an appropriate firewall type for the system.

Cryptographic Key Management. Cryptography ensures the integrity, confidentiality, authenticity and non-reputability in digital communications. The security patterns play the vital role for the selection and proper implementation of cryptographic algorithms.

4.4 Pattern Documentation

We present the pattern documentation format developed for the Pattern-Oriented Software Architecture series, because this format fits with our research objective and goal. Some of the notable parts of a security pattern are: *Name, Example, Context, Problem, Solution, Structure, Dynamics, Implementation, Example Resolved, Variants, Known Uses, and Consequences.* (Yoshioka, et al., 2008) (Meszaros & Doble, 1997) Some of these important pattern parts are further explained below:

Name. It is the foremost important part of any security pattern. It contains the name and a short summary of the pattern.

Example. This part presents the real-world examples of a problem in order to prove the need for a security pattern.

Context. The context of security pattern describes the setting and conditions of vulnerable scenario. It is helpful for categorising security patterns. Such categorised patterns are required to ensure the total security of the vulnerable business process.

Problem. A problem occurs whenever an asset, such as a system, or an application, is protected in an insufficient way against an attack. This part presents the problem for which pattern proposes the solution.

Solution. Depending on the nature of the problem, pattern proposes the solution for the business process. This solution might suggest the potential modifications – which are required to be modified in one or more different levels of the business process.

Structure. This part presents a detailed design of the structural aspects of a pattern.

Consequences. A description of the benefits and demerits of a solution helps everybody to understand the consequences of applying a security pattern. This part of security pattern is useful to show how the security pattern could be applied in a wrong way. This helps to warn everyone about the hidden pattern application dangers and motivates to choose another variant of the pattern.

4.5 Security Risk-oriented Pattern Template

Table 2 presents the Security Risk-oriented Pattern (SRP) template proposed by (Ahmed & Matulevičius, 2011). The template is prepared by aligning the important parts of security patterns with the ISSRM components. We follow the guideline - proposed by (Ahmed & Matulevičius, 2011), for instantiating this SRP template and use it to document ten Security Risk-oriented Patterns (SRP) for the research.

Table 2 Security Risk-oriented Pattern Template adapted from (Ahmed & Matulevičius, 2011)

Entry		Description
Pattern name		This represents the pattern and its security context. It helps to remember and refer to a particular pattern. Normally, the name of the secured business activity is stated here.
Pattern Description		It describes the potential pattern application scenario. This part includes information regarding the business activity, its input and outputs, and the circumstances in which it is applicable.
Asset-related concepts	Asset	An asset is any valuable element which is necessary in accomplishing the organisation's goal.
	Business asset	A business asset can be the information, processes, or skills essential for business's main operation.
	IS asset	An IS asset supports business asset, and it is a component of IS.
	Security criterion	A security criterion is a constraint on business asset, which is expressed through confidentiality, integrity and availability of business asset.
Risk-related concepts	Risk	A risk is composed of event(s) and their deleterious impacts on one or more assets.
	Impact	An impact is the potential bad consequences of a risk.
	Event	An event is a combination of threat and vulnerability.
	Threat	A threat agent initiates a threat by using attack method to harm one or

		multiple IS assets by exploiting their vulnerabilities.
	Vulnerability	A vulnerability is the weakness or flaw of IS asset.
	Threat agent	A threat agent has means to cause harm to IS assets.
	Attack method	An attack method is the technique using which a threat agent fulfils threat.
Risk treatment related concepts	Risk treatment	A decision such as: avoidance, reduction, retention for risk mitigation.
	Security requirement	Security requirement is the refined form of risk treatment decision.
	Control	A control is the implementation of security requirements.
Related pattern(s)		The place for presenting information about the other related SRPs.

4.6 Summary

The chapter begins with a brief history of patterns in software engineering and with a definition of security patterns. We come to know about the advantages of using patterns-based security solution in security risk management. This motivates us for pursuing pattern-based security solution for mitigating business process security risks. In addition, the chapter focuses on the domains where patterns are being used, and we discover that we are already being benefited by utilising the patterns in enterprise security and risk management domain. The essential parts needed to document a security pattern are mentioned in detail. Above all, the chapter presents the Security Risk-oriented Pattern Template developed by (Ahmed & Matulevičius, 2011) . We use it for developing ten Security Risk-oriented Patterns (SRP) for the research. In the next chapter, we define the scope of SRP development by considering vulnerability classifications.

Chapter 5 Security Risk Classification

There are numerous security risks present in IT field. Developing single Security Risk-oriented Pattern (SRP) to address each of these is a lengthy process. Therefore, before starting to develop the SRPs, we need to define the scope of their application area. We decide to address different security risk categories on the basis of vulnerability taxonomy. The main rationale is: vulnerabilities are the primary origins of security risks, and a group of vulnerabilities belong to the same category are responsible for causing similar risks. In this chapter, various vulnerability taxonomies are discussed, and then one of the taxonomies is selected as a classification paradigm.

5.1 Vulnerability Taxonomies

RISOS (Research Into Secure Operating Systems) project is one of the early studies of computer security and privacy. It proposes seven main categories of operating system security issues (R.Abbott, et al., 1975):

- Incomplete Parameter Validation
- Inconsistent Parameter Validation
- Implicit Sharing of Privilege / Confidential Data
- Asynchronous Validation / Inadequate Serialization
- Inadequate Identification / Authentication / Authorization
- Violable Prohibition / Limit
- Exploitable Logic Error

Later, Landwehr, Bull, McDermott and Choi classify vulnerabilities from three different perspectives. These perspectives are presented in figure 11 (Landwehr, et al., 1994).

Genesis. By what means the problem comes into the system.

Time. In which stage of the production cycle the problem moves into the system.

Location. In which part of the system the problem is apparent.

All these perspectives are later subdivided into different parts. In figure 11, three different colours: green, blue and orange represent three different levels of this classification.

One of the major setbacks of this classification is the inability of classifying several existing vulnerabilities. For instance, vulnerability cannot be classified unless one does not know how it has entered into the environment.

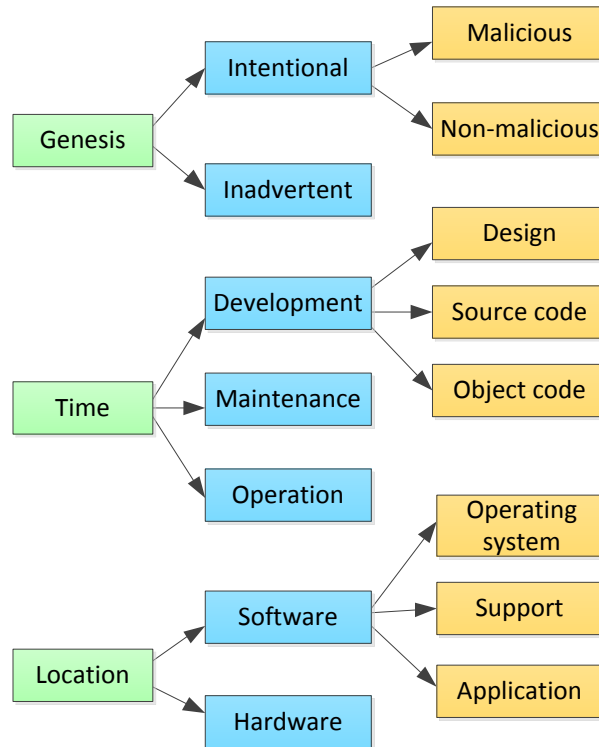


Figure 11 Vulnerability Classification by (Landwehr, et al., 1994)

Tsipenyuk, Chess and McGraw find out that all the above mentioned schemas have several common limitations (Tsipenyuk, et al., 2005). First, the classification becomes ambiguous because of the wide coverage by the categories. Second, the implementation-level and design-level defects are not separately defined in these taxonomies. Third, the taxonomies are not consistent about outlining the categories with respect to the cause of the problem. In order to address all these above mentioned issues, later they suggest their own taxonomy - which is mentioned in ‘*Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors*’ (Tsipenyuk, et al., 2005)

5.2 Seven Pernicious Kingdoms of Vulnerability

Tsipenyuk, Chess and McGraw use two distinct terminologies: *Phylum* and *Kingdom*. *Phylum* is defined as a certain type of coding error. For instance, ‘Buffer Overflow’ represents a phylum. On the other hand, a group of phyla - which share the same theme is considered as *Kingdom*. For example, ‘Errors’ is a Kingdom. The seven kingdoms are presented in Figure 12 by using purple coloured elliptical circles. In the same figure, there is also an additional kingdom represented by pink coloured circle, which covers the issues of software execution environment.

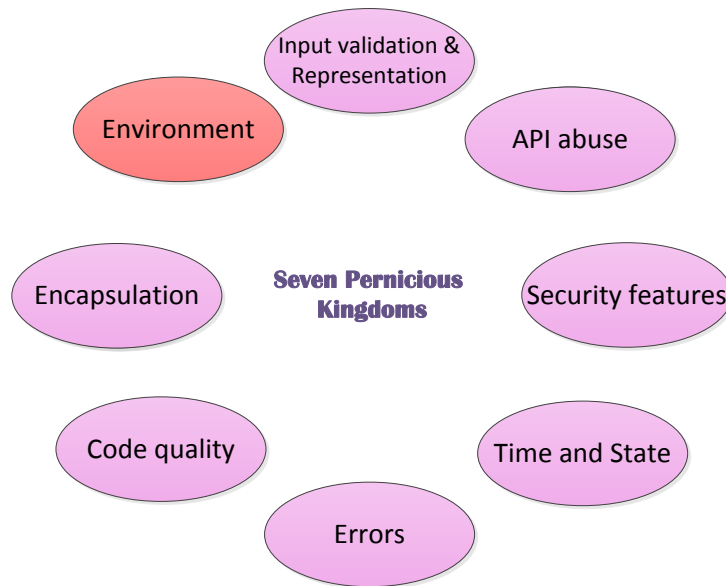


Figure 12 Seven Pernicious Kingdoms of Vulnerability (Tsipenyuk, et al., 2005)

Following is a summary of all vulnerability categories or kingdoms from ‘*Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors*’ (Tsipenyuk, et al., 2005)

Input Validation and Representation. Security issues belonging to this category arise because of trusting input from untrusted domain. XSS attacks, SQL injection, Buffer Overflow are some of the notable examples of this category.

API Abuse. Application Programming Interface or API can be resembled as a joint between a caller and a callee. The abuse can take place when caller calls a callee based on an assumption or expectation - which is not fulfilled. For instance, the return of a non-random value by callee - inside the `SecureRandom` subclass, is an API abuse.

Security Features. Software security does not mean security software. This kingdom mainly discusses about security types such as access control, cryptography, authentication etc.

Time and State. Modern computing is heavily based on distributed computing. It facilitates the sharing of huge amount of computational load by using resource located in different parts. However, this could cause unexpected interactions between concurrent running processes, threads and as well as information. Deadlock, Signal Handling Race Conditions and File Access Race Condition (aka TOCTOU) are some of the phyla belong to this kingdom.

Errors. This kingdom comprises the errors - which occur in software. Most of the time, proper error handling remains absent, and this situation results in leaking out confidential information to unauthorised persons. Possible examples include: Empty Catch Block, Catching NullPointerException etc.

Code Quality. Software programs developed by unskilled programmers suffer from bad quality of source code. This situation paves the way for the intruders to manipulate the behaviour of the programs. Memory leakage, Null deference, Uninitialized variables are some of the phyla included in this kingdom.

Encapsulation. Encapsulation hides one user's data from another user, and creates division between validated and invalidated data. This kingdom deals with the problems such as: Trust boundary violation, System information leak etc.

Environment. Finally, Tsipenyuk, Chess and McGraw create this kingdom to group various security vulnerabilities - which are not directly related to source code. However, considering these vulnerabilities is still important in order to ensure the security of software. Insecure compiler optimization, missing error handling, weak access permissions are couple of the phyla described in this kingdom.

5.3 Summary

This chapter summarises few vulnerability taxonomies and classifications. We discuss vulnerability categories proposed by (R.Abbott, et al., 1975), (Landwehr, et al., 1994) and (Tsipenyuk, et al., 2005). Tsipenyuk, Chess and McGraw mention some of the common limitations of former two taxonomies. These limitations motivate us to select the vulnerability taxonomy model - mentioned in '*Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors*' (Tsipenyuk, et al., 2005), as the taxonomy paradigm for our research. We use this taxonomy to estimate the scope of Security Risk-oriented Patterns (SRP) application area. This estimation is needed in order to develop SRPs - which address different risks arising due to each of the seven vulnerability categories mentioned in the taxonomy paradigm. In the next chapter, we present the SRPs in textual format and also by using BPMN diagrams.

Part II

Contribution

Chapter 6 Security Risk-Oriented Patterns

We present ten Security Risk-oriented Patterns (SRP) in this chapter. Their textual description is given by using Security Risk-oriented Pattern Template proposed by (Ahmed & Matulevičius, 2011). These patterns are modelling language-independent. We use Business Processes Modelling Notation (BPMN) to represent them graphically too. While preparing the diagrams, we use BPMN extensions proposed by (Altuhhova, et al., 2012) to overcome the limitations of BPMN in security analysis. Each Security Risk-oriented Pattern is composed of a textual description and three diagrams: Example business process, Potential threat analysis, and Annotated security requirement. The Example business process diagram represents an instance of a vulnerable business process. The Potential threat analysis diagram explains how an attacker could launch an attack to the vulnerable business process. Lastly, Annotated business process diagram represents the same Example business process diagram, but it is annotated with security requirement – which is proposed by the corresponding SRP.

6.1 SRP1: Securing data that flow between the business entities

E-business heavily depends on transferring data packets from one computer to another through Internet- which is insecure (Otuteye, 2003). Online banking, shopping, ticket purchase are some notable examples of e-business. Companies doing e-business should ensure the confidentiality of client’s private data (Velmurugan, 2009). This security pattern addresses the security risk in online data transmission.

1. Organisational scenario & Security context identification

Pattern name	Securing data that flow between the business entities
Pattern description	This pattern secures the data transmit between the business entities i.e. stakeholders involved in the business process.
Related pattern(s)	No related patterns

2. Asset identification & Security objective determination

Business Asset	Data which is submitted and employed by business
IS Asset	Input interface, Transmission medium that transfers data and business/server
Security criteria	<ul style="list-style-type: none">• Confidentiality of data• Integrity of data

3. Risk analysis & assessment

Risk	An attacker intercepts the transmission medium due to its characteristics to be intercepted and manipulates the data leading to loss of data confidentiality or integrity.
Impact	<ul style="list-style-type: none">• Harm of at least one business asset (i.e. harm of data submitted and stored in the database)• Harm of at least one IS asset (i.e. loss of reliability of the transmission medium)• Negation of security criteria (i.e. negation of data confidentiality and integrity)
Event	An attacker intercepts the transmission medium due to its characteristics to be intercepted and misuses the data due to the lack of crypto-functionality at

Threat	the input interface and server (Tsipenyuk, et al., 2005).
Vulnerability	An attacker intercepts the transmission medium and manipulates the data. <ul style="list-style-type: none"> • Characteristics of transmission medium to be intercepted (Tsipenyuk, et al., 2005). • Lack of crypto-functionality at input interface and server (Tsipenyuk, et al., 2005).
Threat agent	An attacker with means to intercept transmission medium by acting as a proxy
Attack method	<ul style="list-style-type: none"> • Intercept transmission medium by establishing a proxy between input interface and server (Barnum, 2007) (Project, 2009). • Misuse data: <ol style="list-style-type: none"> (a) Capture, modify and pass data to the database. (b) Capture, read and keep data for the later use.

4. Risk treatment & Security requirements

Risk treatment	Risk reduction
Security requirement	<ul style="list-style-type: none"> • Make data unreadable to attackers. (Mitigates the risk of data confidentiality) • Verify the received data with the original. (Mitigates the risk of data integrity)
Control	<ul style="list-style-type: none"> • Cryptographic algorithm • Checksum algorithm
Risk treatment	Risk avoidance
Security requirement	Change the transmission medium which does not have the ability to be intercepted
Control	<ul style="list-style-type: none"> • Client physically delivers the data to company. • Data have to be saved by company's employee.

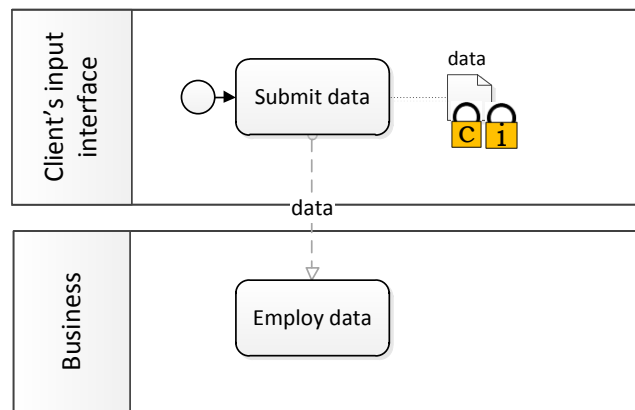


Figure 13 Example business process

In figure 13, a client uses the input interface in order to send data to a business which later employs the data for performing activities in the business process. While travelling through any wired or wireless medium, data can be intercepted by any unauthorised third party (Barnum, 2007).

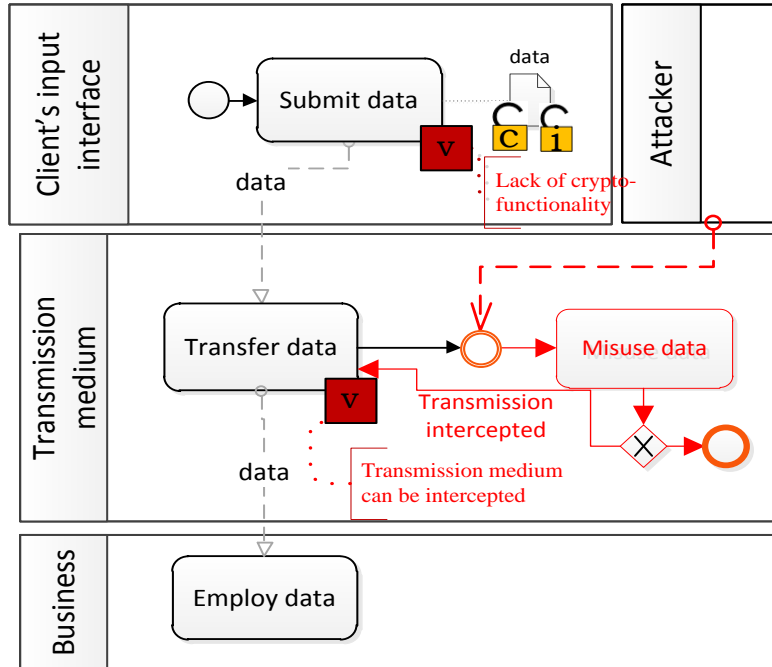


Figure 14 Potential threat analysis

Figure 14 shows the security risk which arises when an attacker intercepts the transmission medium and submits wrong data into the transmission medium again. In figure 15, we annotate the necessary security requirements: 'Make data unreadable to attackers' - which mitigates the risk of confidentiality, and 'Verify the received data with the original' - which mitigates the risk of losing data integrity.

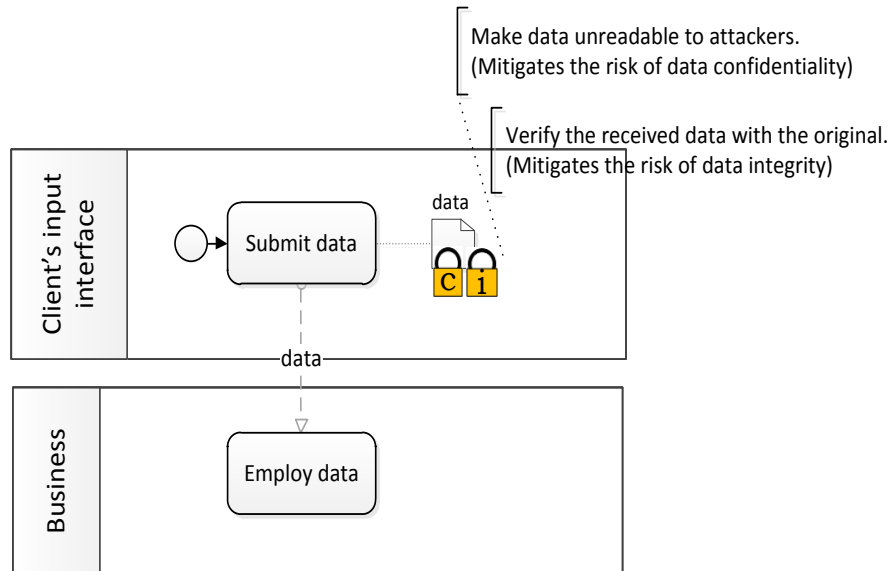


Figure 15 Annotated security requirement

6.2 SRP2: Securing input interface for allowing valid data enter into the business process

The biggest problems in software security exist because software takes inputs from outside (McGraw, 2006) (Gegick & Williams, 2006). When developing software for an IT system, the developer should not trust any data coming from outside into the system (Viega & Messier, 2003). Business enterprises largely suffer from SQL injection (Project, 2011), Cross-site scripting (XSS) (Auger, 2011), XPath Injection (Project, 2009) and various other attacks involving malicious inputs.

1. Organisational scenario & Security context identification

Pattern name	Securing input interface for allowing valid data enter into the business process.
Pattern description	This security pattern ensures data validity and rejects unwanted data when it enters into business process.
Related pattern(s)	No related patterns

2. Asset identification & Security objective determination

Business Asset	Business process which is executed after the data submission.
IS Asset	Input interface
Security criteria	<ul style="list-style-type: none"> • Availability of the business process. • Integrity of the business process.

3. Risk analysis & assessment

Risk	An attacker submits data and malicious script because input interface does not inspect entering inputs thus leading to the compromise of input interface and loss of integrity or/and availability of the business process.
Impact	<ul style="list-style-type: none"> • Integrity of the business activities is broken (e.g. costing business both financially and socially for personal gains). • Input interface is compromised. • Alter business activities because the required activities are not available.
Event	Attacker submits data and malicious script through input interface which doesn't inspect incoming data.
Threat	An attacker submits the data and malicious script.
Vulnerability	Input interface does not inspect data inflow.
Threat agent	An attacker who is capable of writing malicious scripts.
Attack method	<ul style="list-style-type: none"> • Submits data and malicious script. • Malicious script changes the business process either breaking its execution or changing the rules of the process.

4. Risk treatment & Security requirements

Risk treatment	Risk avoidance
Security requirement	Only accept incoming data in predefined format.
Control	<ul style="list-style-type: none"> • Input validation - satisfies the required criteria (Jeremiah & Anton, 2007). • Input sanitization - transforms the input to an acceptable format (Jeremiah & Anton, 2007) • Input filtration - blocks or allows part of input data based on the

acceptable criteria (Jeremiah & Anton, 2007).

- Input canonicalization - converts the input data from possible representations to a standard canonical representation acceptable to the application (Clarke, 2009).

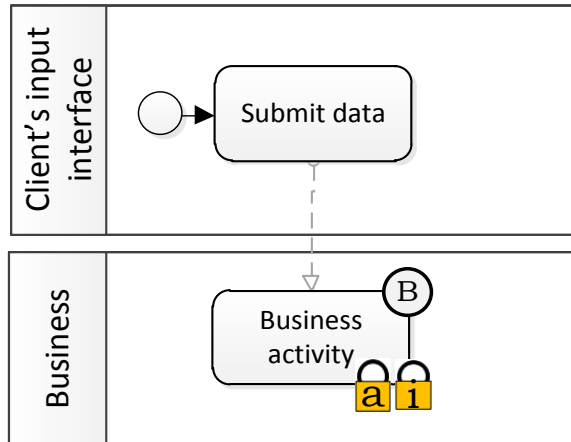


Figure 16 Example business process

In figure 16, the data is flowing from outside into the system. On the basis of this data, the business activity is carried on. Next, figure 17 shows that an attacker is sending malicious data to nullify the integrity and availability of the business activity.

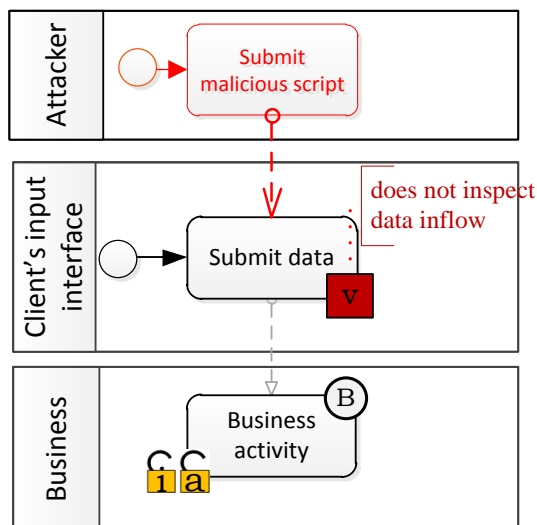


Figure 17 Potential threat analysis

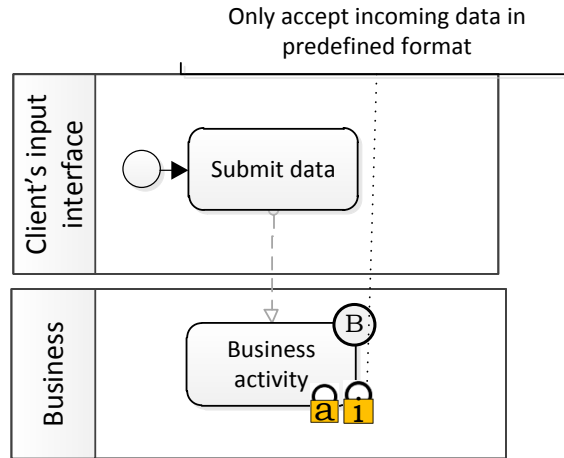


Figure 18 Annotated security requirement

We annotate the figure 18 with security requirement, and propose to accept all incoming data into predefined format - which includes input filtering, sanitizing, and canonicalization.

6.3 SRP3: Protecting the integrity of business activity by securing receiving interface

In earlier days, we could easily verify the identity of the sender of a paper document by examining the signature of the sender. This signature validation property is important in this age of electronic mail too (Rivest, et al., 1978). For ensuring mutual trust in e-business, we should also make sure the actual sender of a document cannot deny that he indeed has sent the digital document. This is often referred as ‘nonrepudiation’ (Security, 2001).

1. Organisational scenario & Security context identification

Pattern name	Protecting the integrity of business activity by securing receiving interface
Pattern description	Companies often make their business decision on the basis of data coming from outside of their own business process. If the data origin is not properly verified, then the decision taken on basis of it may prove wrong. The goal of this security pattern is to protect the integrity of company’s decision making skill by securing receiving interface.
Related pattern(s)	No related pattern

2. Asset identification & Security objective determination

Business Asset	Business activity
IS Asset	Receiving interface
Security criteria	Integrity of business activity

3. Risk analysis & assessment

Risk	A company faces financial loss because of implementing incorrect strategies based on falsified digital document - which comes inside into business
------	--

	process through its receiving interface - which does not verify sender's identity.
Impact	<ul style="list-style-type: none"> • The integrity of decision making skill becomes compromised. • Receiving interface becomes target of future attacks. • Company selects wrong strategy which fails to generate revenue. • As a result, it faces financial loss.
Event	An attacker sends false digital document through company's receiving interface - which does not verify the document sender's identity.
Threat	By pretending to be a genuine sender, an attacker sends fallacious digital document to a company.
Vulnerability	Receiving interface does not verify digital document sender's identity (Pragar & Bingiganavale, 2003)
Threat agent	An attacker who is able to pretend as a genuine sender of digital document.
Attack method	<ul style="list-style-type: none"> • An attacker prepares falsified digital document. • Then, he passes it as an input into a company's internal business process through vulnerable receiving interface.

4. Risk treatment & Security requirements

Risk treatment	Risk reduction
Security requirement	Verify the identity of digital document sender (Shaw, 2001).
Control	Digital signature scheme (Katz, 2007) (Katz, 2010).
Risk treatment	Risk avoidance
Security requirement	Stop exchanging data in electronic format.
Control	Hard copy data bearing sender's signature.

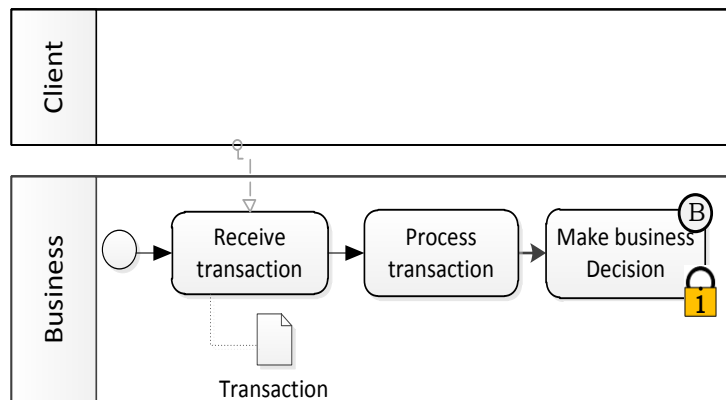


Figure 19 Example business process

In figure 19, we try to illustrate the scenario: an employee of a company is receiving data coming from outside of the company. On the basis of the received data, later he takes business decision.

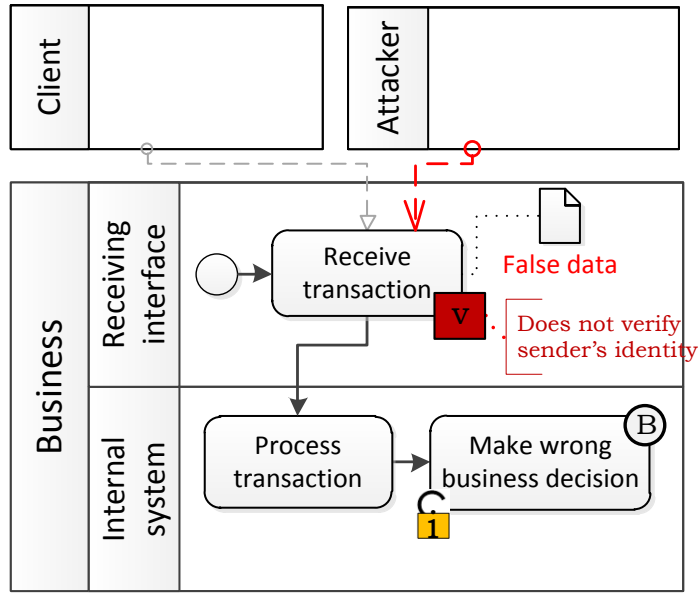


Figure 20 Potential threat analysis

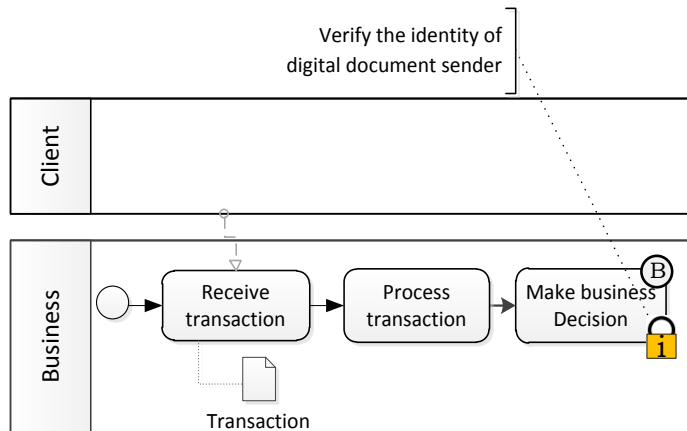


Figure 21 Annotated security requirement

Figure 20 explains when an attacker - disguising himself as a genuine document sender, sends false or manipulated data. Since the receiver assumes that the document is coming from the actual sender - probably by looking at the email domain, later he performs the next business activity on the basis of false data. As a result, the company faces financial loss. In figure 21, we annotate the security requirement which suggests for addressing this security risk by the verification of sender's digital signature at the receiving interface.

6.4 SRP4: Protecting IS from Denial Of Service (DOS) attack

Denial of Service (DOS) attack is one of the common major security problems in IT field (Loukas & Oke, 2009) (Zhang, et al., 2010). It occurs when a company's server runs out of resources - which are required in order to offer services to the users (University, 2001). There are many variants of DOS attacks techniques present in IT field (Zhang, et al., 2010) (Eddy, 2007) (Leyden, 2008). This SRP is developed for analysing and addressing these DOS attacks.

1. Organisational scenario & Security context identification	
Pattern name	Protecting IS from Denial Of Service (DOS) attacks
Pattern description	Denial Of Service (DOS) attack prevents legitimate users from using service provided by business and leads to downtime, thus service consumers lose confidence in that business. The goal of this security pattern is to ensure the availability of business service.
Related pattern(s)	No related pattern
2. Asset identification & Security objective determination	
Business Asset	Offered service
IS Asset	Server database
Security criteria	Availability of service
3. Risk analysis & assessment	
Risk	An attacker performs DOS attack i.e. causes the offered service become unavailable to the users, through exploiting intentionally created half-open (Baccala, 1997) connections by sending false messages to the server database - which allows unlimited number of TCP connections.
Impact	<ul style="list-style-type: none"> • Availability of the service is compromised • In some cases, server database runs out of memory, crashes, or becomes inoperative.
Event	Attacker exploits intentionally created half-open connections by sending false messages to the server database - which allows unlimited number of TCP connections.
Threat	Attacker can exploit intentionally created half-open connections by sending false messages to the server database.
Vulnerability	Server TCP implementation allows unlimited number of connections (CISCO, n.d.).
Threat agent	An attacker capable of initiating new connections in a faster rate than the victim system can fulfil the pending connections.
Attack method	Attacker uses client software which creates too many half-open connections (Baccala, 1997) to the victim server.
4. Risk treatment & Security requirements	
Risk treatment	Risk reduction
Security requirement	Install IP filtering to restrict internal incoming and outgoing packets.
Control	Proper router configuration.

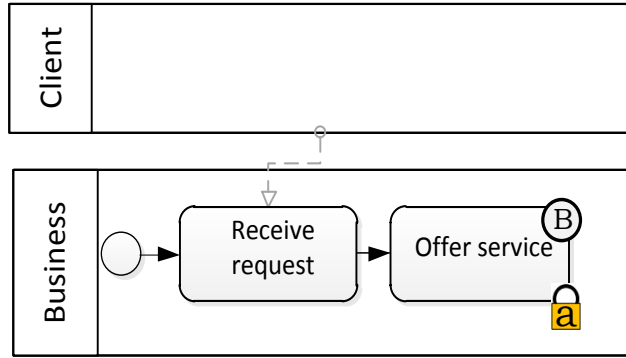


Figure 22 Example business process

In figure 22, a business is accepting outside request and offering services in reply. It could be online banking service, medical service, ticket booking service etc. All these types of services are desired to be available all the time. However, figure 23 shows how an attacker intentionally keeps requesting the same service multiple times by opening numerous half-open connections (Baccala, 1997) at the same time.

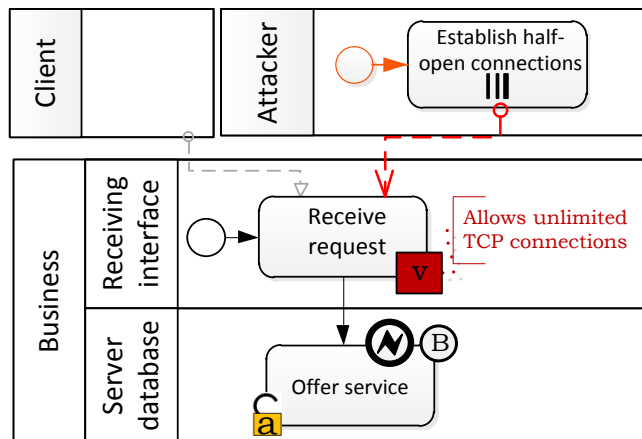


Figure 23 Potential threat analysis

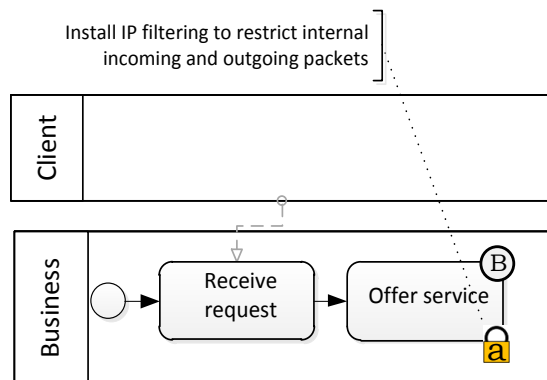


Figure 24 Annotated security requirement

Most of the time, resource limitation imposes the restriction on server for accepting limited number of connections. As a result too many half-open connections are likely to cause the system to become unresponsive, which negates the constant service provided by the company server. Finally we add the necessary security requirement in figure 24.

6.5 SRP5: Applying multilevel access rights to retrieval interface

Employees often need to access data from company database in order to perform daily jobs. On the other hand, companies also need to protect their confidential information from unauthorised person; otherwise they might face financial risk (Smith, 2006). As a solution, in many companies, especially in military services (Stamp & Hushyar, 2006), the data are categorised into different levels according to different types of access rights.

1. Organisational scenario & Security context identification	
Pattern name	Applying multilevel access rights to retrieval interface
Pattern description	This patterns describes how data can be protected from misuse by using Multi-Level Security (MLS) (Anderson, 2008)
Related pattern(s)	No related pattern
2. Asset identification & Security objective determination	
Business Asset	Data
IS Asset	Retrieval interface
Security criteria	Confidentiality of data
3. Risk analysis & assessment	
Risk	A company's confidential data fall into the hand of an unauthorised person because one of its internal employees has access to the interface - which displays all the data without any restrictions.
Impact	<ul style="list-style-type: none"> Confidentiality of data is negated. Retrieval interface becomes target of future attacks. Data can be forwarded to another unauthorised person.
Event	An employee accesses company data through its retrieval interface - which does not support data access protocol and retrieves confidential data.
Threat	An employee accesses company data through its retrieval interface and retrieves confidential data.
Vulnerability	Retrieval interface does not have data access protocol.
Threat agent	An employee unauthorised to read confidential data.
Attack method	<ul style="list-style-type: none"> Launch the retrieval interface. Retrieve available data from database.
4. Risk treatment & Security requirements	
Risk treatment	Risk reduction
Security requirement	Restrict access to confidential data and provide access to relevant data (Bell, 2005)



Figure 25 Example business process

Figure 25 is presenting a data retrieving scenario by an employee. Figure 26 shows a rouge employee is accessing confidential data - which he or she should not read, using the retrieval interface. The retrieve request is being executed in the database and the retrieved data is sent back to wicked employee (i.e. attacker).

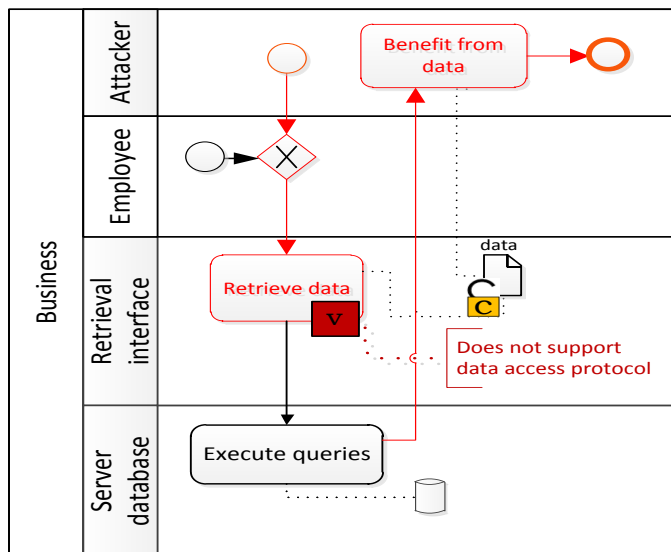


Figure 26 Potential threat analysis

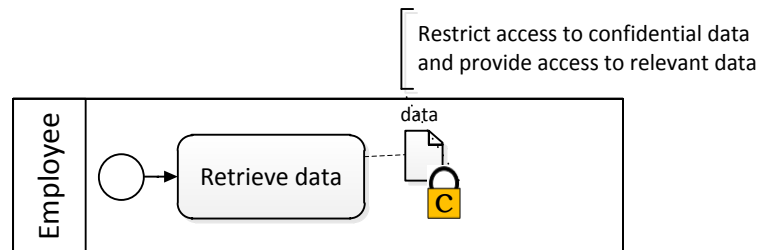


Figure 27 Annotated security requirement

In order to avoid this security risk, the retrieval interface should check every data access request and grant access according to the corresponding employee's data access right. This security requirement is annotated in figure 27.

6.6 SRP6: Securing data confidentiality from unauthorised person in a data store

Attacker can use malicious inputs (e.g. SQL scripts, malicious XPath (Project, 2009)) to divulge confidential data from an enterprises system database (Sen, 2007) (Ragan, 2011). He might use the secret data for stealing money, performing social engineering or even become financially benefited by passing it to a third party.

1. Organisational scenario & Security context identification	
Pattern name	Securing data confidentiality from unauthorised person in a data store
Pattern description	This security pattern secures a data store by storing confidential data in encrypted format (Security, 2002)
Related pattern(s)	No related pattern
2. Asset identification & Security objective determination	
Business Asset	Data
IS Asset	Data store
Security criteria	Confidentiality of data
3. Risk analysis & assessment	
Risk	A company faces financial loss because of losing confidential information acquired from unencrypted data stolen from its data store by an attacker through SQL command execution (McMillan, 2011)
Impact	<ul style="list-style-type: none"> The confidentiality of data becomes compromised. Data store becomes the target for more future attack. <p>This impact could provoke the following impacts:</p> <ul style="list-style-type: none"> Attacker can pass company's secret data to its opponents in order to be financially benefited. Client and stakeholders lose confidence on company's privacy policy. Company may even face financial loss in future.
Event	An attacker reveals information from retrieved data - which was stored in unencrypted format into data store, by SQL command execution.
Threat	An attacker retrieves data from data store by executing SQL commands.
Vulnerability	Confidential parts of data are not encrypted in data store.
Threat agent	An attacker who has or acquired unauthorised access to data store.
Attack method	Retrieve data from data store by executing SQL command (Project, 2011).
4. Risk treatment & Security requirements	
Risk treatment	Risk reduction
Security requirement	Make confidential data invisible to unauthorised person. (Corporation, 2001)
Control	<ul style="list-style-type: none"> Use of cryptographic algorithm (Knudsen, 1998) Use of hashing algorithm
Risk treatment	Risk avoidance
Security requirement	Do not store confidential data into data store.

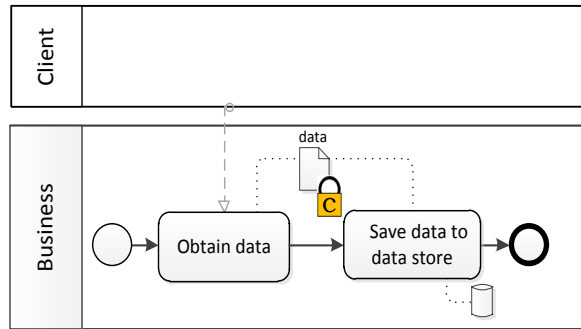


Figure 28 Example business process

In figure 28, a business process is acquiring data from outside and saving it to the database. In case an attacker (figure 29) becomes successful in stealing the secret data from the database by SQL injection, or XPath injection, he might be able read and interpret the confidential information (e.g. credit card numbers, password, username, address etc.) (Lemon, 2008) (Keizer, 2008).

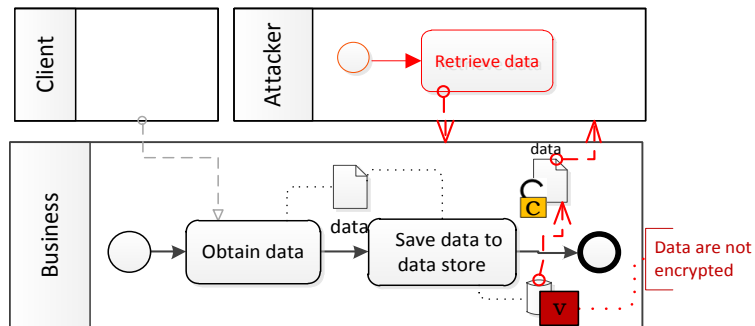


Figure 29 Potential threat analysis

So, in order to be in the safe side, it is advisable to encrypt confidential data before storing it to a database. We have annotated this security requirement in figure 30. If this requirement is met, then even if an attacker becomes successful stealing secret data, he will not be able to decipher it.

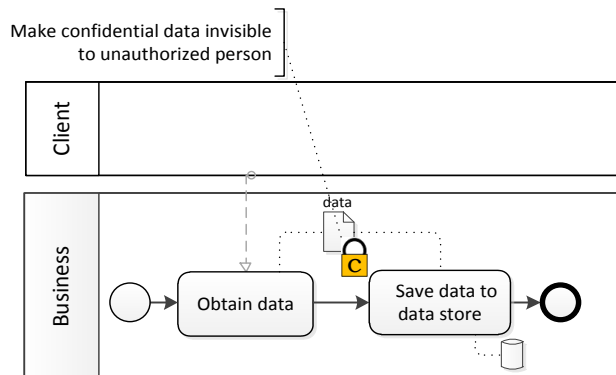


Figure 30 Annotated security requirement

6.7 SRP7: Securing business activity from deadlock condition

‘Deadlock’ condition represents a situation when two concurrent running processes wait for each other to be finished (Padua, 2011). In case of long waiting time, the web service provided by an enterprise becomes unavailable to the user. Therefore, before designing any system, the designers should be aware of any possibilities of potential deadlock (Havender, 2010).

1. Organisational scenario & Security context identification	
Pattern name	Securing business activity from deadlock condition
Pattern description	This security pattern avoids the deadlock condition and describes a mechanism to handle multiple requests in parallel.
Related pattern(s)	No related patterns
2. Asset identification & Security objective determination	
Business Asset	Business activity
IS Asset	Server database
Security criteria	Availability of the business activity
3. Risk analysis & assessment	
Risk	An attacker causes the business activity to become unavailable for indefinite time by triggering multiple actions to server database - which does not follow consistent locking discipline for accessing system resource, thus leading to deadlock condition in the system.
Impact	<ul style="list-style-type: none"> • Availability of the business activity is compromised. • Server database crashes due to deadlock condition.
Event	An attacker triggers multiple actions to server database - which does not follow consistent locking discipline for accessing system resource, and leads to deadlock condition in the system.
Threat	An attacker is able to trigger multiple actions which can create deadlock condition into the system.
Vulnerability	Absence of consistent locking discipline in server database (Tsipenyuk, et al., 2005).
Threat agent	Attacker having knowledge of system API and resources (Dalci, 2007)
Attack method	<ul style="list-style-type: none"> • Attacker triggers an action which uses systems resource. • Next, he initiates next action which waits for the former action to complete, but the former action waits for another resource. Thus he creates a ‘hold and wait’ condition. (Coffman, et al., 1971) (Dalci, 2007)
4. Risk treatment & Security requirements	
Risk treatment	Risk reduction
Security requirement	<ul style="list-style-type: none"> • Required processes should request all the resources before starting up. • Required processes should release all their resources before requesting further more resources.
Control	<ul style="list-style-type: none"> • Serialising tokens • All-or-none algorithms
Risk treatment	Risk avoidance

Security requirement
Controls

Advance resource allocation technique for required process.

- Banker's algorithm (Tannenbaum, 1987)
- Wait/Die and Wound/Wait algorithms (Özsu & Valduriez, 2011) utilizes symmetry-breaking technique

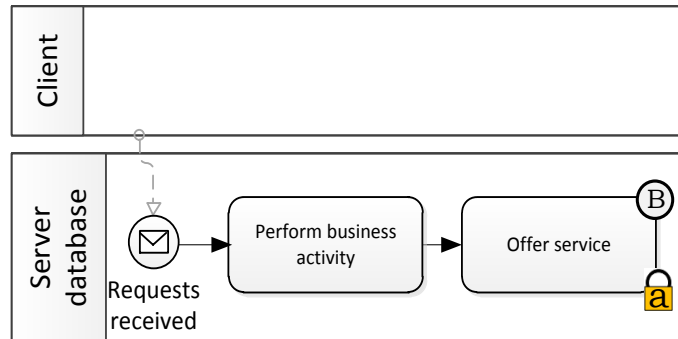


Figure 31 Example business process

In figure 31, client requests are being received by the server and then it performs actions. In figure 32, an attacker initiates a request and the server is working to fulfil it. While the process of fulfilment is still in progress, the attacker initiates the second request - which waits for the first request to be fulfilled. If the server waits for some other resource to complete the first request, then the waiting time for the second request will be too long, thus the server might experience a deadlock situation.

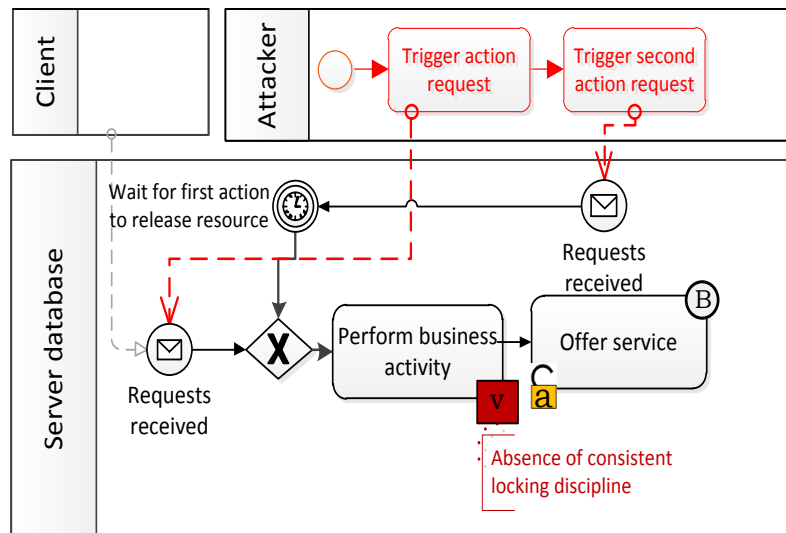


Figure 32 Potential threat analysis

To protect business process from deadlock risk, we mark figure 33 with necessary security requirement. We recommend that each business process activity should request all necessary resources before execution. This will ensure that no process waits unnecessarily for extra system resource.

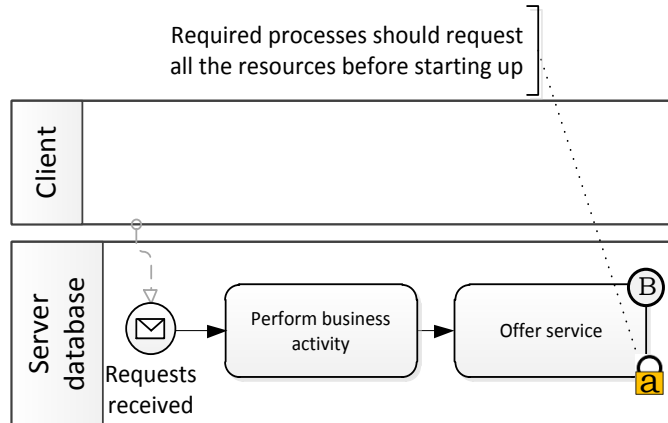


Figure 33 Annotated security requirement

6.8 SRP8: Ensuring atomicity of business transactions to protect data integrity

Process atomicity ensures the separation or isolation from other parallel running processes (MSDN, 2010). An atomic operation can have two possible consequences: either it is successful in changing the state of resource, or in case of failure, it does not have any affect into the resource (Lo, 2005). Transactional business processes such as: Mobile recharge, money transfer, online shopping etc. require process atomicity for ensuring data consistency (Lodde, et al., 2011).

1. Organisational scenario & Security context identification

Pattern name	Ensuring atomicity of business transactions to protect data integrity
Pattern description	This security pattern addresses the atomicity of business transaction in order to ensure that the manipulated data is committed only after a successful transaction.
Related pattern(s)	No related patterns

2. Asset identification & Security objective determination

Business Asset	Business transaction
IS Asset	Server database
Security criteria	Integrity of business transaction

3. Risk analysis & assessment

Risk	A company's business transaction loses its integrity by an attacker who causes business activity failure or causes the activity to abort in abnormal way in a server database which does not have proper resource shutdown or release mechanism or having poor error handling, thus ending up writing conflicting data into database.
Impact	<ul style="list-style-type: none"> • Integrity of the transaction is compromised. • Broker ends up with chained or bundled transactions. • Buyers become forced to perform aggregate transactions. • Optional transactions occur during procurement (Wang & Das, 2001).

Event	An attacker is able to cause business activity failure or causes the activity to abort in abnormal way in a server database which does not have proper resource shutdown or release mechanism or having poor error handling, thus ending up writing conflicting data into database.
Threat	An attacker is able to cause business activity failure or causes the activity to abort in abnormal way in server database.
Vulnerability	<ul style="list-style-type: none"> • Improper Resource Shutdown or Release (Classification, 2012). • Poor error handling (Tsipenyuk, et al., 2005).
Threat agent	An attacker with the ability of consuming, destroying, or disrupting an activity or resource required to perform a normal business transaction.
Attack method	Depending on the nature of the resource the attacker may use these following methods to make the target process unavailable by gaining some privileges on the system: (Classification, 2012) <ul style="list-style-type: none"> • Resource Depletion through Flooding • Resource Depletion through Allocation • Resource Depletion through Leak • Denial of Service through Resource Depletion

4. Risk treatment & Security requirements

Risk treatment	Risk reduction
Security requirement	Process and data should be recovered by implementing an external transaction tracking mechanism (Cobb, 1997).
Control	<ul style="list-style-type: none"> • Proper exception handling. • Implement standard compensation logic.
Risk treatment	Risk reduction
Security requirement	Install IP filtering to filter or restrict internal incoming and outgoing packets.
Controls	Proper router configuration

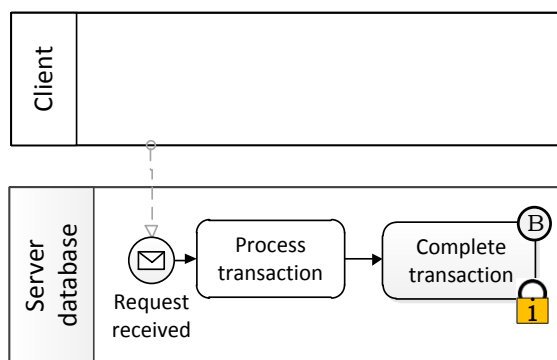


Figure 34 Example business process

Figure 34 represents a common scenario where a web transaction is performed on the basis of received parameters from the system user. In case there is an absence of proper input filtering, the transaction might fail when an attacker supplies invalid parameter values (figure 35) (McGraw, 2006) (Gegick & Williams, 2006). Eventually, that particular transaction process may end up raising an exception and we should

make sure to implement external mechanism - which will help the process to roll-back to its previous stage (figure 36).

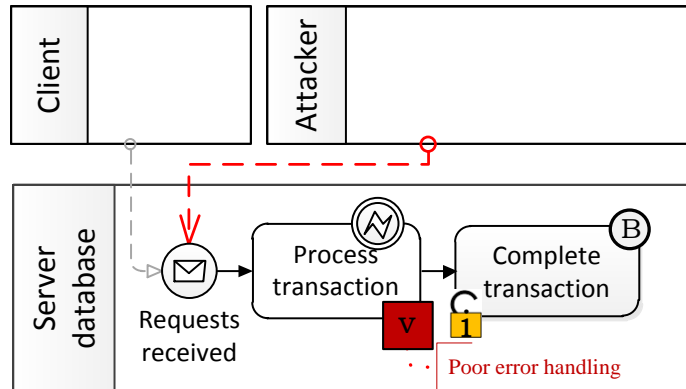


Figure 35 Potential threat analysis

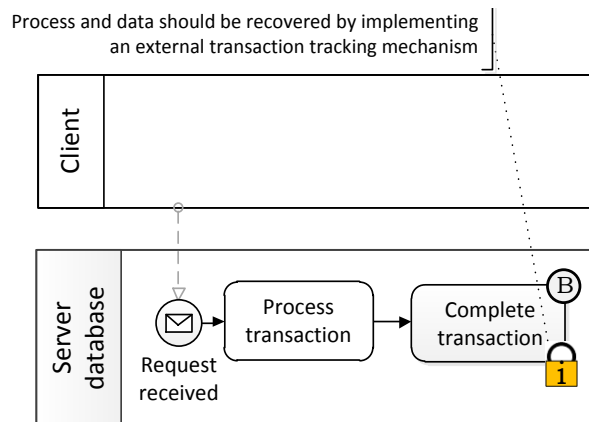


Figure 36 Annotated security requirement

6.9 SRP9: Protecting data integrity in Time Of Check Time Of Use (TOCTOU) situation

One of the conspicuous features of distributed and parallel computing is resource sharing and multi-tasking (Butelle & Coti, 2011). However, this feature can also turn into a system risk (e.g. race condition) if there is no defined mechanism on accessing shared resource by multiple process threads (Chen, 2006).

1. Organisational scenario & Security context identification

Pattern name	Protecting data integrity in Time Of Check Time Of Use (TOCTOU) situation.
--------------	--

Pattern description	This pattern describes how data integrity can be protected in TOCTOU (Bishop & Dilger, 1996) (Enumeration, 2011) (Project, 2009) scenario.
Related pattern(s)	No related pattern
2. Asset identification & Security objective determination	
Business Asset	Data
IS Asset	Server database
Security criteria	Integrity of data
3. Risk analysis & assessment	
Risk	A company's data integrity becomes compromised because of a group of threat agents perform concurrent operations on the same data from different locations to a server database which does not support atomic operation or locking mechanism on data.
Impact	<ul style="list-style-type: none"> • Integrity of data is compromised. • Disruption occurs in server databases normal operation. • Attackers become financially benefited (Library, 2012). • Company faces financial loss (Library, 2012).
Event	A group of separately located threat agents perform parallel manipulations on same data - where the server database does not support atomic operation or locking mechanism on it.
Threat	A group of threat agents concurrently manipulate the same data from different locations.
Vulnerability	<ul style="list-style-type: none"> • System operations are not atomic i.e. there is time gap between resource check and resource usage (Bratus, et al., 2008) • Absence of locking mechanism during data access.
Threat agent	A group of attackers, who have access to same data.
Attack method	<ul style="list-style-type: none"> • Initiate simultaneous sessions. • Perform concurrent data manipulation.
4. Risk treatment & security requirements	
Risk treatment	Risk reduction
Security requirement	Implement locking protocol on data access (Enumeration, 2011).
Control	Lock file mechanism. (Wheeler, 2004)

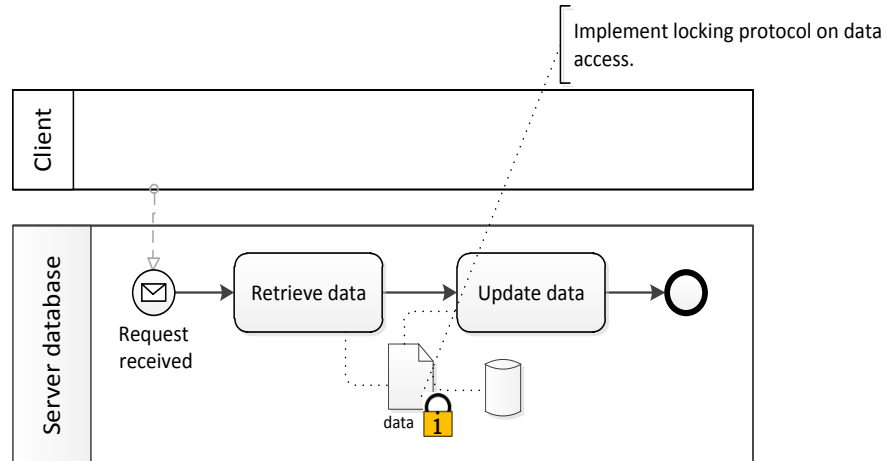


Figure 39 Annotated security requirement

6.10 SRP10: Preventing System Information Leakage

System automation speeds up the process execution and reduces the overall process duration. However, during system failure, the software - which automatically runs the system, passes error reports to the system user. The error reports help the system developers to detect, identify and solve the bugs in the system (Office, 2008). These reports often contain confidential data which can help an attacker to become familiar about internal configuration of the system and devise sophisticated future attacks (Office, 2008) (Enumeration, 2011).

1. Organisational scenario & Security context identification

Pattern name	Preventing System Information Leakage
Pattern description	This pattern describes how to prevent confidential server and database information from attacker when an exception is raised in the system.
Related pattern(s)	No related pattern

2. Asset identification & Security objective determination

Business Asset	System internal information
IS Asset	Server database
Security criteria	Confidentiality of system internal information

3. Risk analysis & assessment

Risk	An attacker becomes capable of launching sophisticated attacks by acquiring confidential system internal information gained through intentionally raised exception in a vulnerable server. (Tsipenyuk, et al., 2005)
Impact	<ul style="list-style-type: none"> Confidentiality of system internal information is lost. Acquired information can be used to launch more sophisticated future attacks against server database.

Event	An attacker gains system internal information by generating runtime exception by providing invalid inputs to a vulnerable server which is not properly configured for smart error handling.
Threat	An attacker tries to gain system internal information by generating exception.
Vulnerability	<ul style="list-style-type: none"> • Absence of exception handling techniques or detailed error handling in servlets. (Project, 2007) • Server is not properly configured for exception management. (Auger, 2010)
Threat agent	An attacker who searches for system internal information.
Attack method	<ul style="list-style-type: none"> • Uses input interface in order to establish session with company server. • Tries to generate exception by providing invalid inputs.

4. Risk treatment & Security requirements

Risk treatment	Risk reduction
Security requirement	Address error and exception wisely.
Control	<ul style="list-style-type: none"> • Servlet source code should implement adequate unexpected exception handling. (Papa, 2012) • Perform static code review. • Configure server for using custom error pages. (Support, 2006) (Shaw, 2010)

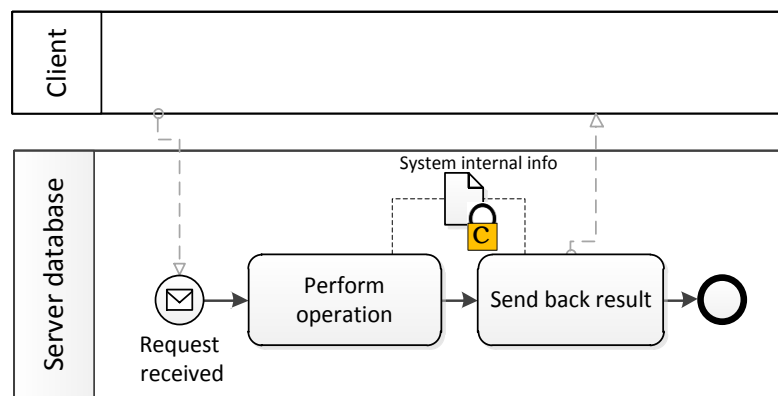


Figure 40 Example business process

Figure 40 shows a part of a web service process. Here, a server of a company receives a request from outside client, performs some operations using the parameters received in the request, and sends back the results to the request origin.

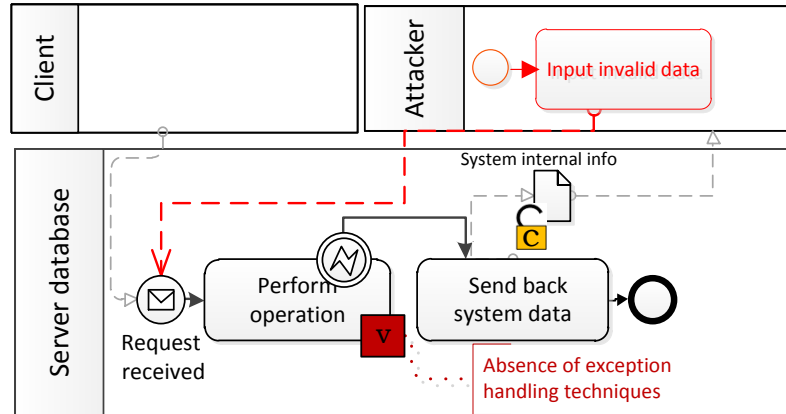


Figure 41 Potential threat analysis

In figure 41, an attacker passes malicious input as a request to the server. The request is successfully received, however while performing the operation - which depends on the input, the process raises an exception and fails to execute further. Finally, since, there was no proper exception handling mechanism present, the system sends back the confidential debugging information to the user of the web service i.e. the attacker.

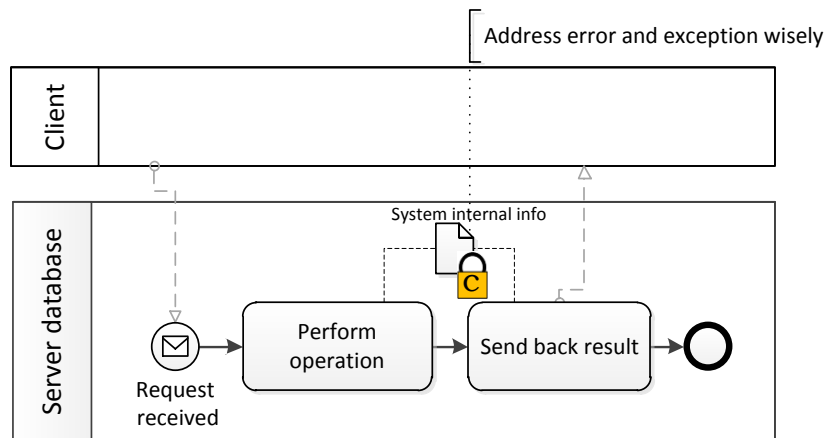


Figure 42 Annotated security requirement

In figure 42, we suggest to address system error and exception carefully.

6.11 Summary

In this chapter, we become familiar with ten Security Risk-oriented Patterns (SRP). SRPs are associated with textual and graphical description for the facilitation of their usability. Later, these patterns are used in order to identify the presence of risks in the business process models. SRPs also help to define appropriate security countermeasures.

Chapter 7 Pattern Application

In previous chapter, we presented ten Security Risk-oriented Patterns (SRP). However, still we do not know how we can use these. In this chapter, we propose the guidelines for applying SRPs in business processes. The guideline consists of four major steps (figure 43): Occurrence identification, Security criterion annotation, Security risk requirement annotation, and Security requirement rationalisation.

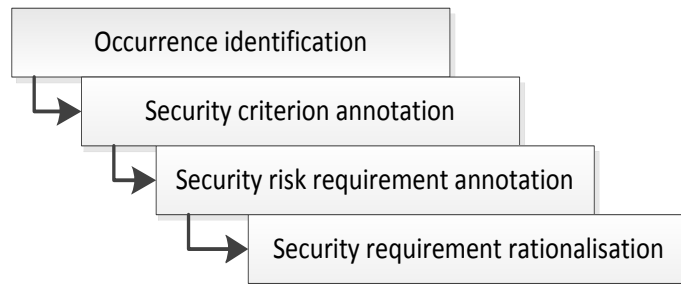
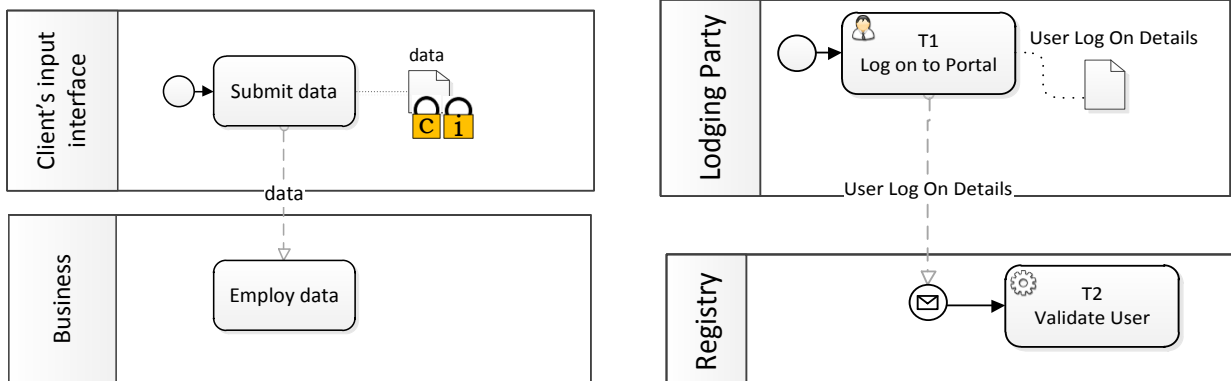


Figure 43 Steps of SRP application guideline

7.1 Step One: Occurrence identification

In the first step, we manually identify the occurrence of SRP in a business process model. This action requires good understanding of process domain and security risk. In figure 44, we show a matched part of a case study business process model (Figure 44 (b)) and the example business process diagram of SRP1 (Figure 44 (a)). A short comparative analysis reveals the correspondence between 'Submit data' (Figure 44 (a)) and 'T1 Log on to portal' (Figure 44 (b)) activities. A good rationale could be: both of these activities deal with data - which enter using input interfaces. Similarly, we also discover the correspondence between tasks 'Employ data' (Figure 44 (a)) and 'T2 Validate user' (Figure 44 (b)), because both of these tasks use data after getting it from input interfaces.



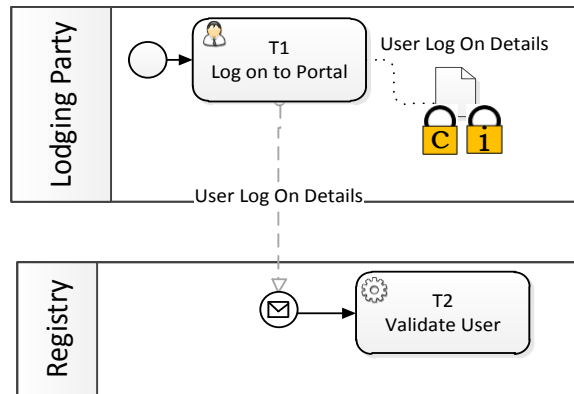
(a) Example business process of SRP1

(b) Part of a case study business process model

Figure 44 Step 1 Occurrence Identification

7.2 Step Two: Security criterion annotation

After finding a matched occurrence from the case study, we identify the vulnerable asset and annotate it with the security criterion. We get the security criterion from the ‘example business process’ diagram of SRP1. In this case, ‘User Log on Details’ is annotated with lock signs (Figure 45) to emphasise that its confidentiality and integrity require protection.

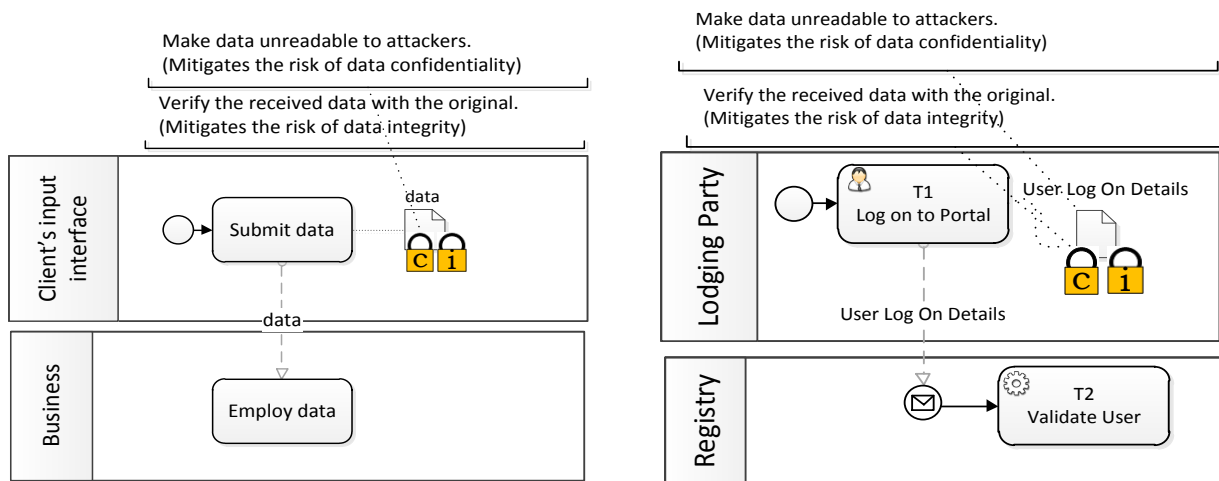


Case study process model is marked with security criteria

Figure 45 Step 2 Security criterion annotation

7.3 Step Three: Security risk requirement annotation

Next, we use BPMN annotation stencil to annotate security requirements (Figure 46). We get the security requirement from the ‘annotated security requirement’ diagram of SRP1. If we have more than one security risk requirement, we have to mark both of them. This creates an option for the decision makers to choose between two available choices. Alternatively, we could also use the combinations of task, event, and gateway constructs to complete this step (Altuhhova, et al., 2012).

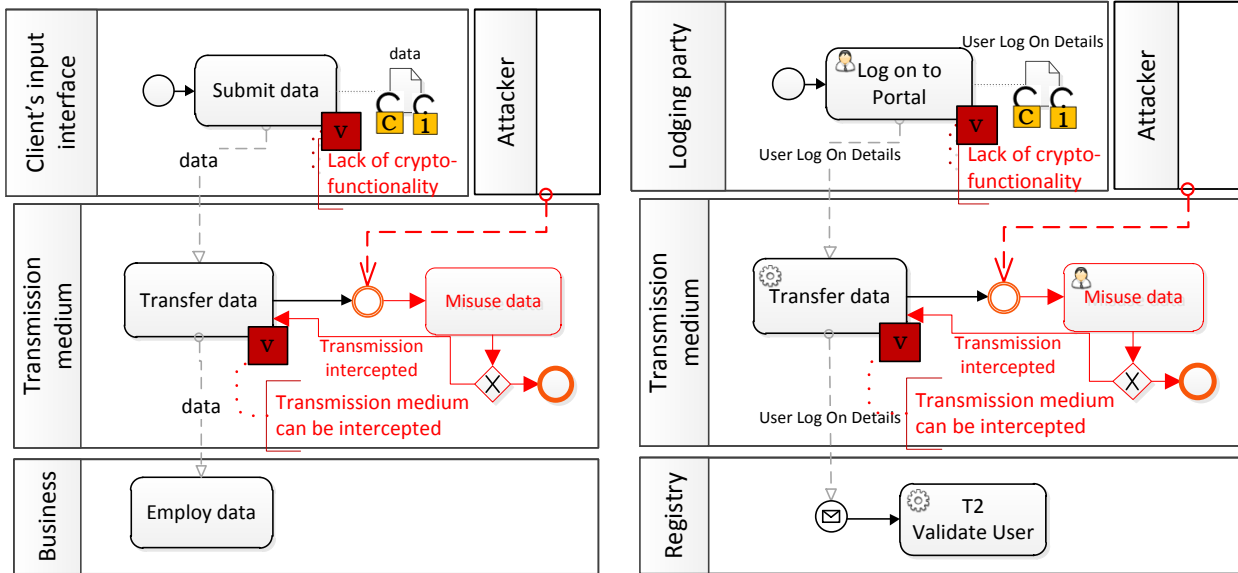


(a) Annotated security requirement diagram of SRP1 (b) Case study process model annotated with security requirement

Figure 46 Step 3 Security risk requirement annotation

7.4 Step Four: Security requirement rationalisation

Finally, we use the potential threat analysis diagram of the SRP1 to show what security breach could occur if one does not fulfil the proposed security risk requirement (Figure 47). This helps the decision makers to decide which one of the security requirements is more important and needs to be satisfied before the other one.



(a) Potential threat analysis diagram of SRP1

(b) Threat analysis in case study process model

Figure 47 Step 4 Security requirement rationalisation

7.5 Summary

In this chapter, we present the guideline for applying Security Risk-oriented Patterns (SRP) in business processes. We elaborately describe the steps of SRP application guideline which consists of four steps. First, we demonstrate how we are able to identify a particular SRP occurrence in a business process model with the help of example business process diagram of that particular SRP. We pinpoint the matches between the SRP and the given business process model, and mention the reasons for considering it as a SRP occurrence. Second, we identify valuable asset and annotate it with security criterion to emphasise its need for protection. Third, we use the security requirement proposed by the SRP to annotate the business process model. The annotation suggests the decision makers to meet the security requirement. Finally, we perform threat analysis with the help of potential threat analysis figure of SRP and inform the decision makers about possible security breach. This threat analysis stands as the rationale behind security requirement proposed by SRP. In the next chapter, we use this SRP application guideline to find SRP occurrences in two case studies and validate all SRPs.

Part III

Validation

Chapter 8 Validation

In this chapter, we validate Security Risk-oriented Patterns (SRP) by conducting two case studies. The case studies involve two business processes collected from two business companies. These companies perform different types of functions and also they are not located at same location. We chose these different companies for the case study in order to examine the usability of SRPs in diverse business processes running in different contexts.

8.1 Experiment Questions

In each of the case studies, we answer the following three experiment questions on the basis of validation results.

1. *Are the security patterns usable?*
2. *Do the patterns exist in real scenario?*
3. *How many risks are found?*

8.2 Validation Methodology

The validation methodology consists of five steps (figure 48). Step 1 – 4 are performed separately in each of the case studies. Step 5 (figure 48, grey coloured box) is performed jointly for both case studies.

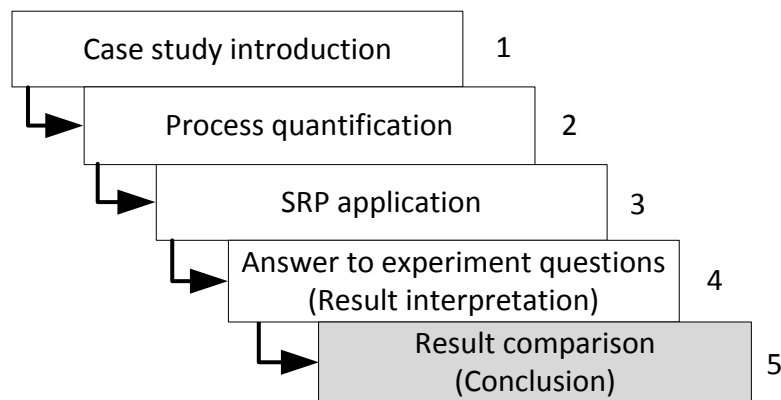


Figure 48 Validation Methodology

In the first step, we provide a short description of the company whose business process is used in case study. In second step, we count the number of processes, sub-processes, events, gateways, pool etc. and present the results in a table. This step helps to understand the extent and complexity of case study business process model. As third step, we apply SRPs using the pattern application guideline described in chapter 7. We also record their occurrences in a second separate table. The number of SRP occurrences is a good indicator of the presence of security risks in business process. In step four, we answer the three predefined experiment questions on the basis of the data - which we get in first and second steps. Finally, in step five, we compare the results acquired from both case studies and present our conclusion.

8.3 Case Study I

8.3.1 Case Study Introduction

The company which is used in the first case study is a government statutory authority which acts as a source of land and property related information for individual, business and Government. This authority is in charge of maintaining land information of a part of Australia. This information includes land maps, satellite pictures, aerial photographs and various other spatial related data. This agency has been maintaining its website for many years. The information seekers can receive their desired information by using this website.

8.3.2 Process Quantification

We manually count the number of processes, sub-processes, events, gateways, pools, tasks, messages and sequence flows present in case study I business process model and present these in Table 3 by using separate columns.

Table 3 Quantitative description of Case Study I business process model

Processes	Sub-process	Events	Gateways	Pools	Tasks	Message flows	Sequence flows
9	73	109	83	68	186	129	492

8.3.3 SRP Application

We apply ten SRPs in case study I business process using the four-step pattern application guideline presented in chapter 7. In table 4, 1st column lists the IDs of SRPs, 2nd column presents the numbers of SRP occurrences in case study I, and finally, 3rd column shows the percentages of the numbers of SRP occurrences in case study I.

Table 4 SRP occurrences in business process of Case Study I

SRP ID	Number of SRP occurrences in case study I business process	An extend, at which pattern influences the business process model
SRP1	33	16.8%
SRP2	33	16.8%
SRP3	25	12.8%
SRP4	35	17.9%
SRP5	39	19.9%
SRP6	6	3.1%
SRP7	12	6.1%
SRP8	1	0.5%
SRP9	4	2.0%
SRP10	8	4.1%
Total	196	100%

8.3.4 Answers to Experiment Questions

EQ1: It is hard to find 100% matches between SRP example business process diagrams and the parts of a case study business process diagram. Therefore, in addition to match the diagrams, we also consider finding the similarities in their contexts and scenarios. These similarities help to find SRP occurrences in case study I (see table 4, column 2) - which shows the usability of security risk-oriented patterns (SRP).

EQ2: The business process which is used in case study I is taken from a statutory authority of Australian Government. This authority has been using this business process for several years to provide land and property related information to individual, business and Government. Table 4 Column 2 is displaying the occurrences of ten SRPs - which shows the existence of our security risk-oriented patterns in real scenario.

EQ3: We find total 196 SRP occurrences in case study I (Table 4, bottom cell of column 2), and because each SRP addresses a security risk, so we are able to identify 196 security risks from case study I.

8.4 Case Study II

8.4.1 Case Study Introduction

Case study II is based on an e-governance application processing system, which is currently being used in an administration bureau of a province in China. In this process, an apartment developer submits an application in order to get the approval for selling commercial apartments. The application proceeds through different checks conducted by different government agencies such as: Property management office, Demolition resettlement verification authority and many other public auxiliary offices. Finally, the process reaches to an end with three different possible decisions regarding developer's application: Approved, Re-Apply, and Rejected.

8.4.2 Process Quantification

Similar to case study I, here we also manually count the number of processes, sub-processes, events, gateways, pools, tasks, message and sequence flows present in case study II business process and present these in Table 5 using separate columns.

Table 5 Quantitative description of Case Study II business process model

Processes	Sub-process	Events	Gateways	Pools	Tasks	Message flows	Sequence flows
17	0	34	119	0	168	0	377

8.4.3 SRP Application

We apply ten SRPs in case study II business process using the four-step pattern application guideline presented in chapter 7. In table 6, 1st column lists the IDs of SRPs, 2nd column presents the numbers of SRP occurrences in case study II, and finally, 3rd column shows the percentages of the numbers of SRP occurrences in case study II.

Table 6 SRP occurrences in business process of Case Study II

Pattern ID	No# of pattern occurrence in business model	An extend, at which pattern influences the business process model
SRP1	6	20.6%
SRP2	6	20.6%
SRP3	8	27.5%
SRP4	4	13.7%
SRP5	2	6.8%
SRP6	3	10.3%
SRP7	0	0%
SRP8	0	0%
SRP9	0	0%
SRP10	0	0%
Total	29	100%

8.4.4 Answers to Experiment Questions

EQ1: In case study II, we consider finding matches in BPMN diagrams, contexts and scenarios between SRP example business processes and the parts of case study business process diagram. After completing analysis, we present the numbers of the SRP occurrences in table 6 column 2. These numbers act as an indicator of the usability of security risk-oriented patterns.

EQ2: The case study II is based on a business process which is used by a provincial Government in China. From table 6 column 2, we see the numbers of SRP occurrences in case study II and, these numbers help to realise the SRPs existences in real scenario.

EQ3: We find total 29 SRP occurrences in case study II (Table 6, bottom cell of column 2), and because each SRP addresses a security risk, so we are able to identify 29 security risks in case study II.

8.5 Threats to Validity

We follow some conventions while performing step 2 and 3 of validation methodology. They are listed below:

- We do not consider any process model which is not prepared using English language.

- Some activity names in process models are unclear, i.e. they only convey proper meanings to the actual process owners or users, but not to us. Therefore, we keep such activities outside of validation scope.
- We do not apply SRPs in the high level process models, because normally these high level process models are composed of sub-processes. Instead, we apply SRPs in their inside processes.

If we do not follow these conventions or try to address the above mentioned factors in different ways, we might expect variations in the numbers of SRP occurrences. Above all, these occurrences are subjective, so it can vary from analysis to analysis.

8.6 Result Comparison

- From table 3 and table 5, we get clear idea about the size and complexity of these two case study business processes. Case study I has total 492 sequence flows, and on the other hand, case study II has total 377 sequence flows. Comparing the number of sequence flows, we can estimate that the case study I is broader than case study II.
- Although larger in size, case study I seems less complicated than case study II. Case study I has less number of gateways than case study II, and these gateways are responsible for increasing the complexity of process model. From table 5, it is also evident that there is not a single sub-process and message flow present in case study II.
- After applying developed security risk-oriented patterns (SRP), we identify 196 pattern occurrences in case study I and 29 patterns occurrences in case study II. While performing the analysis, we notice the process diagrams in case study II lacks details about the process, for instance, there are no separate pools, lanes, sub-processes, as well as no message flows. This could be the potential reason behind the identification of less number of security risks from it.
- In case study I, we manage to find occurrences of all SRPs. In contrast, in case study II, occurrences of SRP7, SRP8, SRP9 and SRP10 are absent.
- Each of the SRPs is developed for the purpose of mitigating single security risk. This notion helps to reach in conclusion that SRPs have detected total (196 + 29) or 225 security risks from both case study I and case study II. Furthermore, we can assume that Case study I is more vulnerable to security risks than case study II.

8.7 Summary

In this chapter, we use two case studies in order to validate security risk-oriented patterns (SRP). We specify three predefined experiment questions and later answer these on the basis of validation results of both case studies. The validation methodology consists of five steps. In each of case studies, we provide a short description of the company to familiarise the readers to the purpose and functions of business process. We use SRP application guideline - mentioned in chapter 7, to find the occurrences of SRPs. Ten of these occurrences (one for each SRP) are presented in the Appendix section of the report. The results of the analysis present quantitative descriptions of the business processes and show the number of occurrences of SRPs in those business processes. In addition, in table 4 and 6, we calculate the percentages of SRP occurrences with the purpose of getting clear idea regarding the extent of SRP

influences on each of the case studies. SRP occurrence identification task depends on security analyst's wise judgement in application context and scenario. We mention a number of conventions - which we follow while performing validation steps. These conventions would help other validators to acquire validation result similar to ours. Finally, we analyse raw data from the tables and present comparative interpretations. Following chapter comes with our final opinions and conclusions regarding this research.

Part IV

Conclusion

Chapter 9 Conclusion

At the beginning of the report, we present our motivation, scope, research questions and research methodology for developing business process in a secured way using pattern-based solution. We discuss about various security risk management frameworks such as: AURUM (Ekelhart, et al., 2009), CORAS (Lund, et al., 2011) and ISSRM (Dubois, et al., 2010), and provide our arguments for choosing ISSRM as the framework of our research along with the detail description of it. Superiority of BPMN (OMG, 2012) over other business process modelling languages is shown by presenting its fulfilment of modelling criteria, and ISSRM - BPMN alignment is established. We introduce the concept of patterns, the advantages of using these, different parts of patterns, and most importantly, present a template - which is aligned with ISSRM, for future SRP development. To determine the extent of SRP application area and to comprehend risk-vulnerability relationship, we scrutinise several vulnerability classifications and end up choosing the one presented in 'Seven Pernicious Kingdoms' as our vulnerability taxonomy paradigm. With our unified knowledge of ISSRM, BPMN, Security pattern and Vulnerability classification model, we develop ten Security Risk-oriented Patterns (SRP). In addition, four-step SRP application guideline is proposed in this report. To demonstrate the effectiveness of SRPs in pattern-based development of secure business processes, we show their usability in two case studies.

9.1 Answer to Research Questions

The primary research question of this thesis was:

RQ 1: *How to make business process secure?*

Answer: In general, a business process analyst collaborates with a security analyst to search for security solutions for business process. At an early stage of process development, the business analyst shares the business process model with a security analyst. He helps the security analyst to identify and comprehend the value of business assets to the company. Both of them also help each other to define the security criterion for the identified business assets. Next, security analyst starts security risk analysis by finding out which IS assets are responsible for supporting the identified business assets. According to ISSRM domain model, vulnerabilities are the characteristics of IS assets, and these are also related to the security risks. Therefore, by identifying IS assets, security analyst is able to trace the origin of potential security risks present in business process model. Next, he follows the Security Risk-oriented Pattern application guideline in order to find the SRPs occurrences in the process model. If SRPs occurrences are detected, security analyst annotates the business process model with the security requirements proposed by the SRPs. Then, he hands over the annotated business process model to the business analyst. This model should provide the business analyst with the complete security needs for his company's business process. He might ask for additional information (i.e. the controls specified by SRPs) from the security analyst regarding the technique of satisfying these proposed security requirements. However, before satisfying any security requirement, the business analyst needs to take into account the costs and time associated with it. So, the proposed security requirements need to be prioritised according to the severity of negative impacts of related risks on the business process. More detail feedback such as: threat analysis diagrams

could facilitate this prioritising task. Finally, by satisfying the security requirements, business analyst develops pattern-based secure business process.

RQ 2: *What are the Security Risk-oriented Patterns to secure business processes?*

Answer: We developed ten Security Risk-oriented Patterns to secure business processes. They are briefly described below.

SRP1 Pattern ensures the integrity and confidentiality of transmitted data between business entities.

It addresses the security risk of unsecured data transmission when two or more business entities exchange data to carry on their business operations. The threat scenario identifies that attacker can intercept the transmission medium which can result into the loss of data confidentiality and integrity. To minimise risk, pattern proposes two security requirements: make data unreadable before transmission, to keep it confidential, and calculate check-sum value, to ensure data integrity. To avoid risk, this pattern proposes to change the transmission medium which cannot be intercepted.

SRP2 Pattern rejects malicious data and ensures the entrance of valid data into system.

It analyses the danger of invalid data which originates from the business clients and enters into the IS of a company. The risk analysis identifies that it can cause the loss of business process integrity and attacker can make business entity unavailable to its clients. To avoid risk, this pattern proposes the requirement which defines a structured format for all incoming data, and restricts data which disregards any predefined format.

SRP3 Pattern verifies the origin of received data and protects the integrity of business decision.

It focuses two issues which can compromise the integrity of business process: the legitimacy and the non-repudiation property of received data. Pattern captures the risks of selecting wrong business strategies, and incorrect initiation of a business process (e.g. process invalid purchase order) - which can take place if a business does not verify sender's verification. To reduce the risk, this pattern introduces the requirement of verifying sender's digital signature.

SRP4 Pattern protects the IS from Denial Of Service (DOS) attack.

It discusses the problem of business service unavailability. An attacker can make a service inaccessible and prevent the legitimate users from using it – which affects a business. To avoid this situation, this pattern proposes the requirement to restrict internal and external packets for a specific time period by using proper router configuration.

SRP5 Pattern implements Multi-Level Security (MLS) in data access and protects data from misuse by attackers.

It addresses the issue of data retrieval interface where unauthorised individuals have access to confidential data. This raises the risk of leaking confidential data which can be misused in order to cause harm to business. To reduce the risk, this pattern proposes security requirement of using Multi-level Security (MLS), which means establishing levels of data access rights, restricting anonymous access at retrieval interface, and keeping track of data retrieval.

SRP6 Pattern saves data from attacker by encrypting it in data store.

It catches the data store issues which stores data in plain format. If an attacker manages to establish access to this kind of data store, he can read confidential data. This compromises the confidentiality of data; and any misuse of secret information can have negative impacts on business. Pattern proposes two solutions to this problem: first, reducing risk by storing confidential data in invisible format (e.g. encrypted); second, avoiding risk by getting confidential data directly from client when it is needed, instead of storing it in data store.

SRP7 Pattern ensures proper handling of parallel requests and protects the IS from deadlock condition.

It describes a deadlock situation where a business activity or service holds a resource for infinite time and requests for the same resource again which creates a resource access lock. An attacker could deliberately create such scenario and make it unavailable to its users. To reduce the risk, this pattern proposes: the processes should request all needed resources in advance or release all of these before requesting any new resource.

SRP8 Pattern maintains the integrity of business transaction by ensuring process atomicity.

It solves data inconsistency problem in a business transaction. A transaction is complete after successful executions of several activities but, a single activity can cause the transaction to abort abnormally- which could result in writing conflicting data into system. This incident harms data integrity and causes business process to malfunction. Pattern proposes to implement external mechanism which tracks the transaction, and calls the compensation logic in case of activity failure to undo all changes which take place before the occurrence of failure.

SRP9 Pattern secures shared data from corruption in TimeOfCheck / TimeOfUse (TOCTOU).

It addresses the problem related to concurrent operations. When multiple activities from different locations access same data at the same time, this situation can cause the loss of data integrity which could lead to business process malfunction. To reduce the risk, the pattern proposes to implement locking protocol on data accessibility.

SRP10 Pattern prevents internal system information from leakage during process exception.

It focuses on the problem of information leakage which happens due to run-time exception. If exception is not properly handled, an attacker can intentionally raise the exception in order to get internal systems information for example, application configuration information - which he is able to mine and devise sophisticated attack to the system. To reduce the risk, this pattern proposes to handle system errors and exceptions wisely, so that the system does not expose internal information to unauthorised user.

RQ 3: *How do the Security Risk-oriented Patterns (SRP) help to secure business processes?*

Answer: In this research, we presented ten Security Risk-oriented Patterns (SRP). We apply SRPs by performing four steps: Occurrence identification, Security criterion annotation, Security risk requirement annotation, and Security requirement rationalisation. In order to help security analysts to perform these

steps, the SRP application guideline is describe in detail in chapter 7. During validation, we discover total 225 SRP occurrences in two industrial business process models. Each of these identified SRP occurrences corresponds to a single security risk. Furthermore, the SRPs consist of the description of security risks, vulnerabilities, security requirements and as well as solution controls – which both business and security analysts can use to address identified security risks. By this way, the Security Risk-oriented Patterns (SRP) help to develop secure business processes.

9.2 Limitations

The research contains several limitations:

- Firstly, it is based on BPMN, which is relatively new concept in business process management. The current version BPMN 2.0 was released in January 2011. Although, there are many graphical constructs available which could be used to represent business process, so far no construct can be found to express security risks. To facilitate our research work, we use ISSRM & BPMN alignment proposed by (Altuhhova, et al., 2012).
- While conducting security risk analysis using BPMN, we accepted a certain level of subjectivity. Different security analysts may come up with different diagrams of the same scenario. This fact is also true in this report, but we tried to mitigate the subjectivity through our discussions with the thesis supervisors.
- During validation, we only consider the business processes prepared in BPMN. Other business processes drawn in other modelling languages (e.g. EPC, UML) are not considered, thus limiting the scope of the thesis.
- The validation process depends on the completeness of a given business process. The business process diagram which we have in case study I is more elaborately drawn than that of case study II, and we assume that due to this reason, we are able to find more security risks in the first case study than the second one.

In the security risk-oriented patterns (SRP), we only propose the controls which can be implemented to prevent risk. We do not provide any detail description regarding control implementation, because we assume the IT engineers of a company possess sufficient knowledge and capability for completing this step.

9.3 Future Work

One of the promising future works related to this research could be the introduction of automation in security risk analysis. Software prototype tool could be developed. It would be able to take a business process model drawn in BPMN as input, and deliver result with annotated security requirements in the model. Developing and representing security risk patterns using other modelling languages can also become a new research direction. We could also calculate the complexity increase in business process after implementing the control. This could be a good decisive factor on whether a specific security requirement might be met. Measuring the performance of security analysts in risk mitigation using SRP is also an important future challenge.

Abstract eesti

Turvaliste äriprotsesside muustritel põhinev arendamine

Naiad Hossain Khan

Magistritöö

Iga andmeturbest huvitatud äriettevõtte valib iseendale sobilikud turvameetmed, et vältida ootamatuid sündmusi ja õnnetusi. Nende turvameetmete esmane ülesanne on kaitsta selle äriettevõtte ressursse ja varasid. Äriettevõttes aset leidvad õnnetused (vähemtähtsad või katastroofilised) on enamikel juhtudel oma olemuselt sarnased ning põhjustatud sarnaste turvariskide poolt. Paljudel andmeturbe spetsialistidel on raskusi leidmaks õiget lahendust konkreetsetele probleemidele, kuna eelmiste samalaadsete probleemide lahendused ei ole korrektselt dokumenteeritud. Selles kontekstis on turvalisuse muustrid (Security Patterns) kasulikud, kuna nad esitavad tõestatud lahendusi spetsiifiliste probleemide jaoks.

Käesolevas väitekirjas arendasime välja kümme turvariskidele suunatud muustrit (SRP ehk Security Risk-oriented Patterns) ja defineerisime, kuidas kasutada neid mustreid vastumeetmetena turvariskidele äriprotsesside mudelite sees. Oma olemuselt on need muustrid sõltumatud modelleerimiskeelest. Lihtsustamaks nende rakendamist, on mudelid esitatud graafilises vormingus äriprotsesside modelleerimise keeles (BPMN).

Me demonstreerime turvariskidele suunatud muustrite (SRP) kasutatavust kahe tööstusettevõtte ärimudeli näite põhjal. Esitame muustrite rakendamise kohta kvantitatiivsed analüüsid ja näitame, kuidas turvariskidele suunatud muustrid (SRP) aitavad demonstreerida andmeturbe nõrku kohti ärimudelites ning pakume välja lahendusi andmeturvalisusega seotud probleemidele.

Selle uurimistöö tulemused võivad julgustada andmeturvalisusega tegelevaid analüütikuid jälgima muustritel-põhinevaid lähenemisi oma äriettevõtete kaitsmiseks, et aidata seeläbi kaasa ka infosüsteemide (Information Systems (IS)) kaitsmisele.

Bibliography

Ahmed, N. & Matulevičius, R., 2011. *A Template of Security Risk Patterns for Business Process*. s.l., s.n.

Ahmed, N., Matulevičius, R. & Khan, N. H., 2012. *Eliciting Security Requirements for Business Process Using Patterns*. s.l., s.n.

Altuhhova, O., Matulevičius, R. & Ahmed, N., 2012. *Towards Definition of Secure Business Processes*. s.l., s.n.

Anderson, R., 2008. *Security Engineering A Guide to Building Dependable Distributed Systems*. 1 ed. Indianapolis: Wiley Publishing.

Auger, R., 2010. [Online]

Available at: <http://projects.webappsec.org/w/page/13246936/Information%20Leakage>

Auger, R., 2011. [Online]

Available at: <http://projects.webappsec.org/w/page/13246920/Cross%20Site%20Scripting>

Baccala, B., 1997. [Online]

Available at: <http://www.freesoft.org/CIE/Course/Section4/10.htm>

Barnum, S., 2007. [Online]

Available at: <http://capec.mitre.org/data/definitions/94.html>

Beckwith, R. W., Vanfleet, W. M. & MacLaren, L., 2004. High Assurance Security Safety for Deeply Embedded, Real time Systems. *Embedded Systems Conference*, June.

Bell, D. E., 2005. Looking Back at the Bell-La Padula Model. *ACSAC '05 Proceedings of the 21st Annual Computer Security Applications Conference*, December. pp. 337-351.

Bishop, M. & Dilger, M., 1996. Checking for Race Conditions in File Accesses. *Computing Systems*, 9(2), pp. 131-152.

Bratus, S., D'Cunha, N., Sparks, E. & Smith, S. W., 2008. TOCTOU, Traps, and Trusted Computing. *Trust '08 Proceedings of the 1st international conference on Trusted Computing and Trust in Information Technologies: Trusted Computing - Challenges and Applications*, January. pp. 14-32.

Butelle, F. & Coti, C., 2011. A Model for Coherent Distributed Memory For Race Condition Detection. *IEEE International Parallel & Distributed Processing Symposium*, September. pp. 584-590.

Chen, L. T., 2006. [Online]

Available at: <http://developers.sun.com/solaris/articles/raceconditions.html>

CISCO, n.d. [Online]

Available at:

http://www.cisco.com/en/US/docs/security/asa/asa82/configuration/guide/conns_connlimits.html

Clarke, J., 2009. *SQL Injection Attacks and Defence*. 1 ed. s.l.:Syngress Publishing.

Classification, C. C. A. P. E. a., 2012. [Online]

Available at: <http://capec.mitre.org/data/definitions/119.html>

Cobb, E. E., 1997. The impact of object technology on commercial transaction processing. *The VLDB Journal — The International Journal on Very Large Data Bases*, August, 6(3), pp. 173-190.

Coffman, E. J., Elphick, M. J. & Shoshani, A., 1971. System Deadlocks. *ACM Computing Surveys*, 3(2), pp. 67-78.

CORAS, 2012. [Online]

Available at: <http://coras.sourceforge.net/>

Corporation, O., 2001. *Database Encryption in Oracle9i*, California: s.n.

Dalci, E., 2007. [Online]

Available at: <http://capec.mitre.org/data/definitions/25.html>

Draw, S., 2012. *Software Design Tutorials*. [Online]

Available at: <http://www.smartdraw.com/resources/tutorials/data-flow-diagrams/#/resources/tutorials/Introduction-to-DFD>

[Accessed 10 May 2012].

Dubois, E., Heymans, P., Mayer, N. & Matulevičius, R., 2010. A Systematic Approach to Define the Domain of Information System Security Risk Management. In: S. Nurcan, C. Salinesi, C. Souveyet & J. Ralyte, eds. *Intentional Perspectives on Information Systems Engineering*. s.l.:Springer-Verlag, pp. 289-306.

Dufresne, T. & Martin, J., 2003. *Process Modeling for E-Business*. s.l., s.n.

Eddy, W., 2007. [Online]

Available at: <http://tools.ietf.org/html/rfc4987>

Ekelhart, A., Fenz, S. & Neubauer, T., 2009. *AURUM: A Framework for Information Security Risk Management*. Hawaii, s.n., pp. 1-10.

Ekelhart, A., Klemen, M. & Weippl, E., 2007. *Security Ontologies: Improving Quantitative Risk Analysis*. Hawaii, s.n., pp. 156a-156a.

Enumeration, C. C. W., 2011. [Online]

Available at: <http://cwe.mitre.org/data/definitions/367.html>

- Enumeration, C. W., 2011. [Online]
Available at: <http://cwe.mitre.org/data/definitions/200.html>
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J., 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. 1 ed. s.l.:Addison Wesley.
- Gaudin, S., 2007. [Online]
Available at: <http://www.informationweek.com/news/199000222>
- Gegick, M. & Williams, L., 2006. An Early Testing and Defense Web Application Framework for Malicious Input Attacks. *ISSRE Supplementary Conference Proceedings*.
- Group, O. M., 2008. [Online]
Available at: <http://bpmi.org/>
- Group, O. M., 2012. [Online]
Available at: <http://omg.org/>
- Havender, J. W., 2010. Avoiding deadlock in multitasking systems. *IBM Systems Journal* , April, 7(2), pp. 74-84.
- IBM, I. B. M. C., 1969. *Flowcharting Techniques*, New York: International Business Machines Corporation (IBM).
- Jakoubi, S., Neubauer, T. & Tjoa, S., 2009. *A roadmap to risk-aware business process management*. s.l., s.n., pp. 23-27.
- Jeremiah, G. & Anton, R., 2007. *XSS Attacks Cross Site Scripting Exploits and Defense*. Burlington: Syngress Publishing.
- Karagiannis, D., Mylopoulos, J. & Schwab, M., 2007. *Business Process-Based Regulation Compliance: The Case of the Sarbanes-Oxley Act*. s.l., s.n.
- Katz, J., 2007. *Introduction to Modern Cryptography: Principles and Protocols*. s.l.:Chpman & Hall/CRC.
- Katz, J., 2010. *Digital Signatures*. 1 ed. s.l.:Springer.
- Keizer, G., 2008. [Online]
Available at:
http://www.computerworld.com/s/article/9080580/Huge_Web_hack_attack_infects_500_000_pages
- Khanmohammadi, K., 2010. *Business Process-Based Information Security Risk Assessment*. s.l., s.n., pp. 199-206.
- Knudsen, J. B., 1998. *Java Cryptography*. 1 ed. s.l.:O'Reilly Media.

Ko, R. K. L., 2009. A computer scientist's introductory guide to business process management (BPM). *Magazine Crossroads*, June.15(4).

Landwehr, C. E., Bull, A. R., Mcdermott, J. P. & Choi, W. S., 1994. A Taxonomy of Computer Program Security Flaws, with Examples. *CM Computing Surveys*, 26(3), pp. 211-254.

Lemon, S., 2008. [Online]

Available at:

http://www.pcworld.com/businesscenter/article/146048/mass_sql_injection_attack_targets_chinese_web_sites.html

Leyden, J., 2008. [Online]

Available at: <http://www.theregister.co.uk/2008/05/21/phlashing/>

Library, i. D., 2012. [Online]

Available at:

<http://developer.apple.com/library/IOs/#documentation/Security/Conceptual/SecureCodingGuide/Articles/RaceConditions.html>

Lodde, A., Schlechter, A., Bauler, P. & Fernand, F., 2011. Data Consistency in Transactional Business Processes. *Perspectives in Business Informatics Research-10th International Conference, BIR 2011, Riga, Latvia*, October.pp. 83-95.

Loukas, G. & Oke, G., 2009. Protection against Denial of Service Attacks: A Survey. *Oxford Journals*, May.53(7).

Lo, Y. M., 2005. *Business process atomicity analysis supporting late task property bindings*, Hong Kong: s.n.

Lund, S., Solhaug, B. & Stølen, K., 2011. *Model-Driven Risk Analysis The CORAS Approach*. 1 ed. s.l.:Springer.

Markoff, J., 2012. [Online]

Available at: http://www.nytimes.com/2012/02/15/technology/researchers-find-flaw-in-an-online-encryption-method.html?_r=3&pagewanted=1&gwh&ref=technology

Mayer, N., 2009. *Model-Based Management of Information System Security Risk*, Belgium: s.n.

McGraw, G., 2006. *Software Security: Building Security In*. 1 ed. s.l.:Addison-Wesley Professional.

McMillan, R., 2011. [Online]

Available at: <http://www.networkworld.com/news/2011/041211-hacker-breaks-into-barracuda-networks.html?hpg1=bn>

Meszaros, G. & Doble, J., 1997. A pattern language for pattern writing. In: *Pattern Languages of Program Design 3*. Boston: Addison-Wesley Longman Publishing Co.

- MSDN, M., 2010. [Online]
Available at: <http://msdn.microsoft.com/en-us/library/aa560115.aspx>
- Office, D. E. – E. S., 2008. [Online]
Available at: http://www.oregon.gov/DAS/EISPD/ESO/docs/ESO_App_Sec_Vulns.pdf?ga=t
- OMG, O. M. G., 2006. *Unified Modeling Language: Infrastructure*, s.l.: OMG.
- OMG, O. M. G., 2012. [Online]
Available at: <http://www.bpmn.org/>
- Otuteye, E., 2003. [Online]
Available at: <http://ausweb.scu.edu.au/aw03/papers/otuteye/paper.html>
- Özsu, M. T. & Valduriez, P., 2011. *Principles of Distributed Database Systems*. 3 ed. s.l.:Springer.
- Padua, D., ed., 2011. *Encyclopedia of Parallel Computing*. Illinois: Springer.
- Papa, J., 2012. [Online]
Available at: <http://msdn.microsoft.com/en-us/magazine/cc164003.aspx>
- Pragar, S. & Bingiganavale, S., 2003. Digital Signature: Application Development Trends in E-Business. *J. Electron. Commerce Res.*, pp. 94-101.
- Project, O. T. O. W. A. S., 2007. [Online]
Available at: https://www.owasp.org/index.php/Top_10_2007-A6
- Project, O. T. O. W. A. S., 2009. [Online]
Available at: https://www.owasp.org/index.php/Man-in-the-middle_attack
- Project, O. T. O. W. A. S., 2009. [Online]
Available at: https://www.owasp.org/index.php/XPATH_Injection
- Project, O. T. O. W. A. S., 2009. [Online]
Available at: https://www.owasp.org/index.php/File_Access_Race_Condition:_TOCTOU
- Project, O. T. O. W. A. S., 2011. [Online]
Available at: https://www.owasp.org/index.php/SQL_Injection
- R.Abbott, et al., 1975. *Security Analysis and Enhancements of Computer Operating Systems*, s.l.: s.n.
- Ragan, S., 2011. [Online]
Available at: <http://www.thetechherald.com/articles/DSLReports-com-breach-exposed-more-than-100-000-accounts/13483/>
- Rivest, R., Shamir, A. & Adleman, L., 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, February, 21(2), pp. 102-126.

- Schmidt, D., Stal, M., Rohnert, H. & Buschmann, F., 2000. *Pattern-Oriented Software Architecture*. 2 ed. s.l.:Wiley & Sons.
- Schumacher, M., 2003. *Security Engineering with Patterns: Origins, Theoretical Models, and New Applications*. 1 ed. s.l.:Springer.
- Schumacher, M. et al., 2004. *Security Patterns Integrating Security and Systems Engineering*. s.l.:s.n.
- Security, R., 2002. *Securing Data at Rest: Developing a Database Encryption Strategy*, s.l.: s.n.
- Security, S., 2001. [Online]
Available at: <http://searchsecurity.techtarget.com/definition/nonrepudiation>
- Sen, R., 2007. [Online]
Available at: <http://www.ibm.com/developerworks/xml/library/x-xpathinjection/index.html>
- Shaw, P., 2001. *E-Business Privacy and Trust: Planning and Management Strategies*. 1 ed. s.l.:John Wiley & Sons.
- Shaw, P., 2010. [Online]
Available at: <http://www.codestyle.org/sitemanager/apache/errors-Custom.shtml>
- Smith, R. E., 2006. Multilevel Security. In: *Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management*. s.l.:Wiley, pp. 972-986.
- Stamp, M. & Hushyar, A., 2006. Multilevel Security Models. In: *Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management*. s.l.:Wiley, pp. 987-997.
- Support, M., 2006. [Online]
Available at: <http://support.microsoft.com/default.aspx?scid=kb;en-us;224070>
- Tannenbaum, A., 1987. *Operating Systems: Design and Implementation*. s.l.:Prentice Hall.
- Tsipenyuk, K., Chess, B. & McGraw, G., 2005. Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors. *IEEE Security & Privacy*, November/December. pp. 81-84.
- University, C. S. E. I. C. M., 2001. [Online]
Available at: http://www.cert.org/tech_tips/denial_of_service.html
- Velmurugan, M., 2009. Security and Trust in E-Business: Problems and Prospects. *International Journal of Electronic Business Management*, 7(3), pp. 151-158.
- Viega, J. & Messier, M., 2003. *Secure Programming Cookbook for C and C++*. s.l.:O'Reilly Media.
- Wang, G. & Das, A., 2001. Models and Protocol Structures for Software Agent Based Complex E-Commerce Transactions. *EC-Web 2001 Proceedings of the Second International Conference on Electronic Commerce and Web Technologies*, pp. 121-132.

Wheeler, D., 2004. *IBM Secure programmer: Prevent race conditions*. [Online]
Available at: <http://www.ibm.com/developerworks/linux/library/l-sprace/index.html>

White, S. A., 2006. *Introduction to BPMN*, s.l.: s.n.

Yoshioka, N., Washizaki, H. & Maruyama, K., 2008. A Survey on Security Patterns. *Progress in Informatics*, Volume 5, pp. 35-47.

Zachman, J. A., 1987. A Framework for Information Systems Architecture. *Turning Points in Computing: 1962-1999*, 38(2/3), p. 454.

Zhang, C., Yin, J., Cai, Z. & Chen, W., 2010. RRED: Robust RED Algorithm to Counter Low-Rate Denial-of-Service Attacks. *IEEE COMMUNICATIONS LETTERS*, May, 40(5), pp. 489-491.

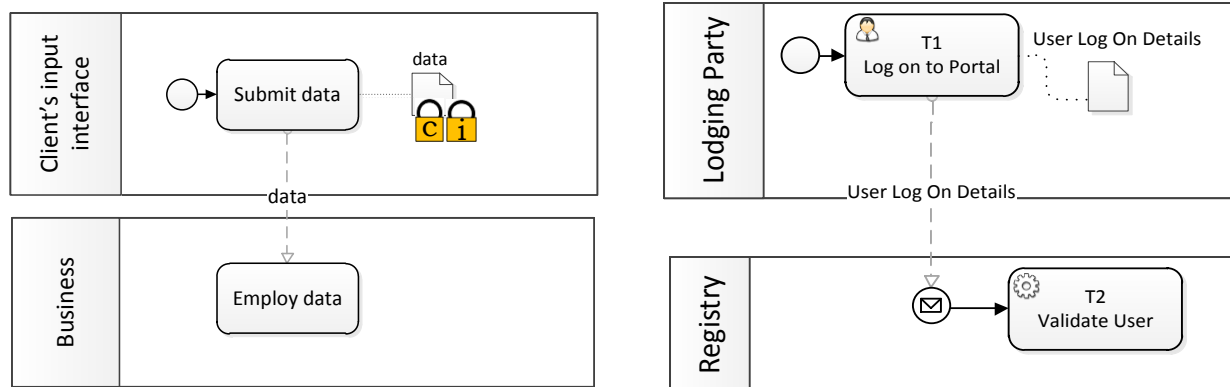
Appendix

In this section, ten occurrences of Security Risk-oriented Patterns (SRP) in Case study I are presented (One occurrence per SRP). We provide short descriptions to demonstrate the matches between Example business process diagrams of each SRP and the parts of case study business process model.

Occurrence of SRP1

In 1.0 Prepare Plan (figure 49 (b)), the Lodging Party is sending *User Log On Details* to the Registry – which validates the details and grants access to the system. However, when data travels through the transmission medium, it could be intercepted by an intruder or attacker. He can steal or manipulate the data and even more, he can retransmit the manipulated data back to the transmission medium. For latter case, the malicious data can cause harm to the Registry online system.

SRP1 (figure 49 (a)) shows *data* is being submitted using the input interface and Business employs the *data* for later usage. The activity *Submit data* can be considered similar to *T1 Log on to Portal* activity, and the *Employ data* can be matched with *T2 Validate User* activity.



(a) Example business process of SRP1

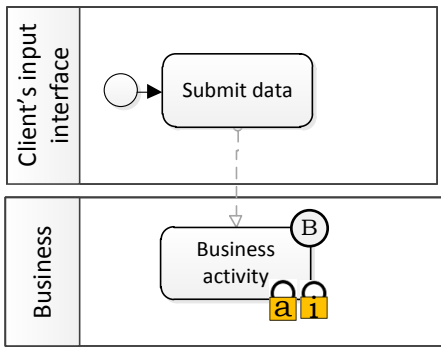
(b) 1.0 Prepare Plan

Figure 49 Occurrence of SRP1

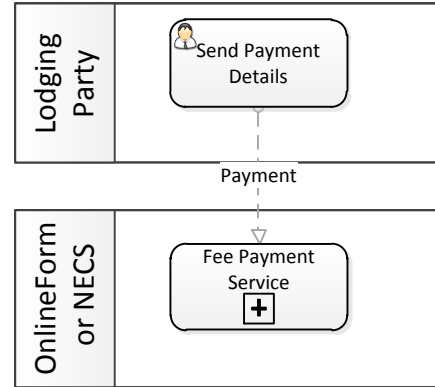
Occurrence of SRP2

In 7.0 Lodge Transaction business process (figure 50 (b)), The Lodging Party is sending *Payment Details* as an input to the Online Form or NECS. This is later being used by the *Fee Payment Service* sub-process activity. If some malicious input enters into the system instead of the *Payment Details*, the *Fee Payment Service*, this malicious input could raise error and cause the activity to crash.

In SRP 2 (figure 50 (a)), the process also depends on the *data* obtained from outside of the system, and we can assume that the *Business activity* is similar to the *Fee Payment Service*.



(a) Example business process of SRP2



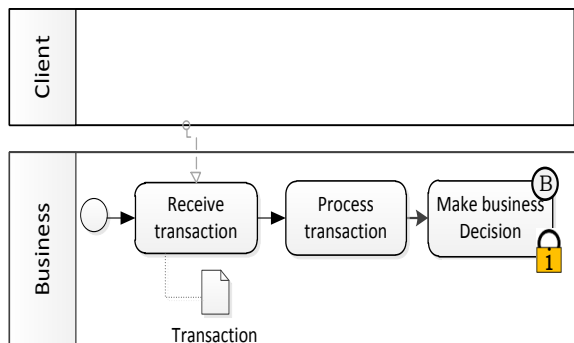
(b) 7.0 Lodge Transactions

Figure 50 Occurrence of SRP2

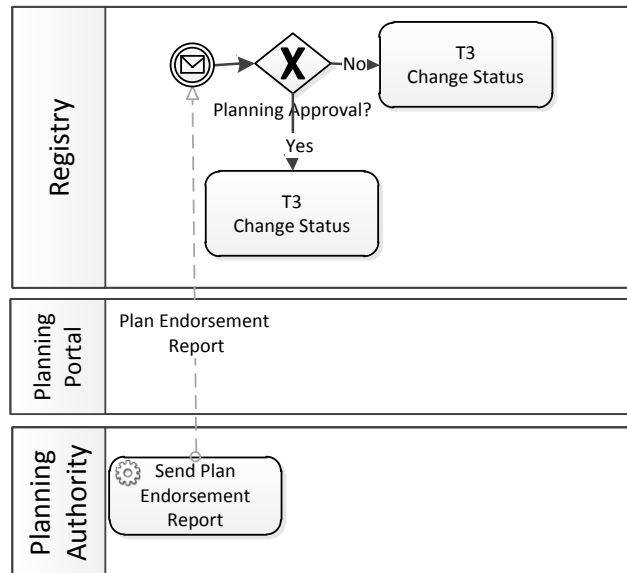
Occurrence of SRP3

In figure 4.0 Approve Plan business process diagram (figure 51 (b)), the Registry receives *Plan Endorsement Report* from the Planning Authority. The report is sent by using Planning Portal. On the basis of the report, the Registry changes the status - either it approves or disapproves. Because the sender of the *Plan Endorsement Report* is not verified upon the receipt, there could be a risk of taking wrong plan approval decision on the basis of unverified data.

In SRP 3 (figure 51 (a)), we show the *Transaction* data is obtained by Business. After obtaining the data, the transaction is processed and at last business decision is taken - which can be aligned with changing the planning approval status in 4.0 Approve Plan business process diagram.



(a) Example business process of SRP 3



(b) 4.0 Approve Plan

Figure 51 Occurrence of SRP3

Occurrence of SRP4

In 1.0 Prepare Plan business process diagram (figure 52 (b)), the Registry is receiving *User Log On Details* from Lodging party and grant access to the system after validating the user details. The Registry provides online service, so there might be a limitation on the number of users who can receive the service at the same time. If an attacker intentionally creates too many half-open connections (Baccala, 1997), this might result into Denial of Service (DOS) (Loukas & Oke, 2009).

In SRP4 (figure 52 (a)), the Business is receiving request and offering services to the user. The server - which is receiving request, might also have a limitation on the number of users who can receive the service. This is a similar case to the Registry in 1.0 Prepare Plan (figure 52 (b)).

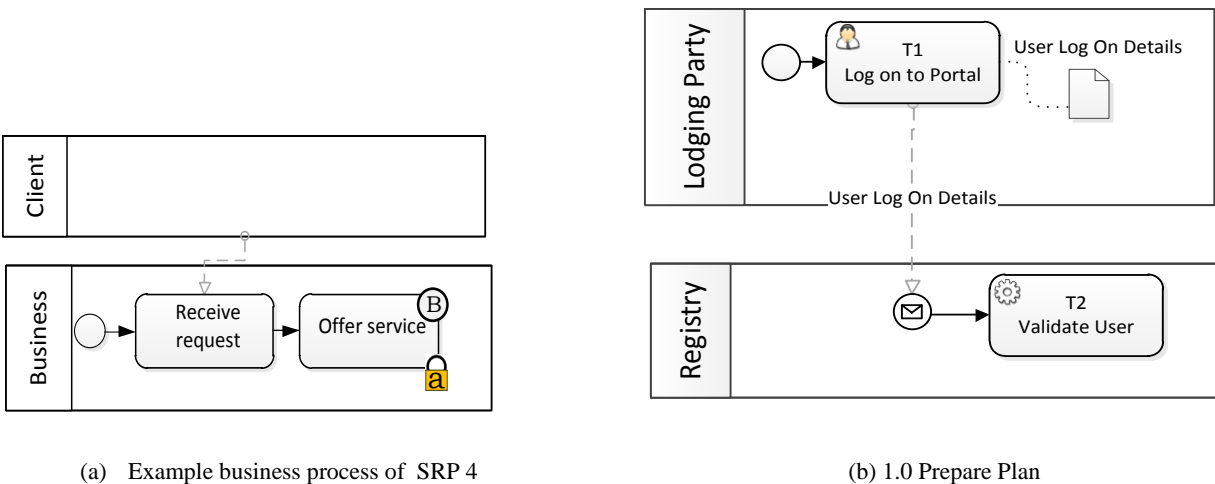


Figure 52 Occurrence of SRP4

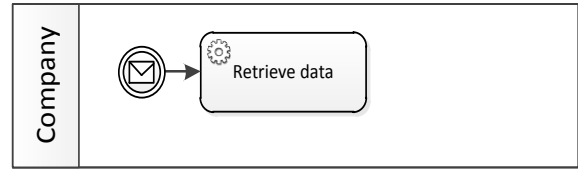
Occurrence of SRP5

In 10.4 Fee Statement Service business process diagram (figure 53 (b)), the company receives a request and then retrieves the data from the database. While accessing the data, the company's employee might reveal confidential information - which he or she should not know.

In SRP5 (figure 53 (a)), Employee retrieves the *data* and without any Multi Level Security (MLS) access protocol (Bell, 2005), the confidentiality of accessed data could be at risk. This scenario is similar to the process scenario presented in 10.4 Fee Statement Service (figure 53 (b))



(a) Example business process of SRP5



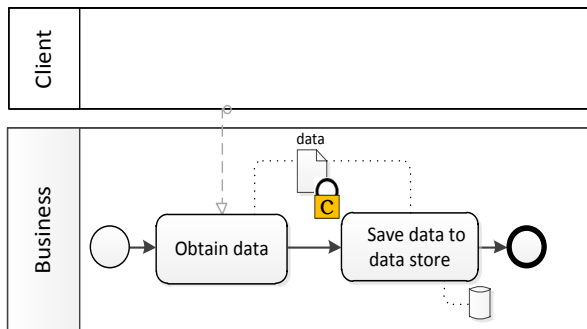
(b) 10.4 Fee Statement Service

Figure 53 Occurrence of SRP5

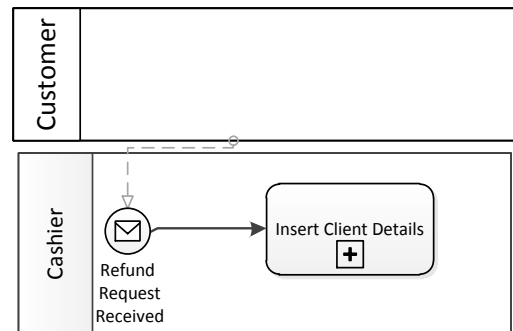
Occurrence of SRP6

In 10.5 Refunds business process diagram (figure 54 (b)), Cashier receives a *Refund request* from customer, and he inserts the client's details into the system. This details can be composed of both confidential (e.g. credit card number, social security number) and non-confidential information. If this confidential information is not encrypted before being saved into a database, it might be susceptible to future security breach. For instance, if an attacker becomes successful in establishing access to the database, he can retrieve all the unencrypted data and acquire the confidential information from it.

In SRP6 (figure 54 (a)), the Business obtains *data* from outside and saves it to the data store. The *Save data to data store* activity is similar to the *Insert Client Details* activity present in the 10.5 Refunds (figure (b)) business process model. This could be a potential match to relate these two business processes.



(a) Example business process of SRP6



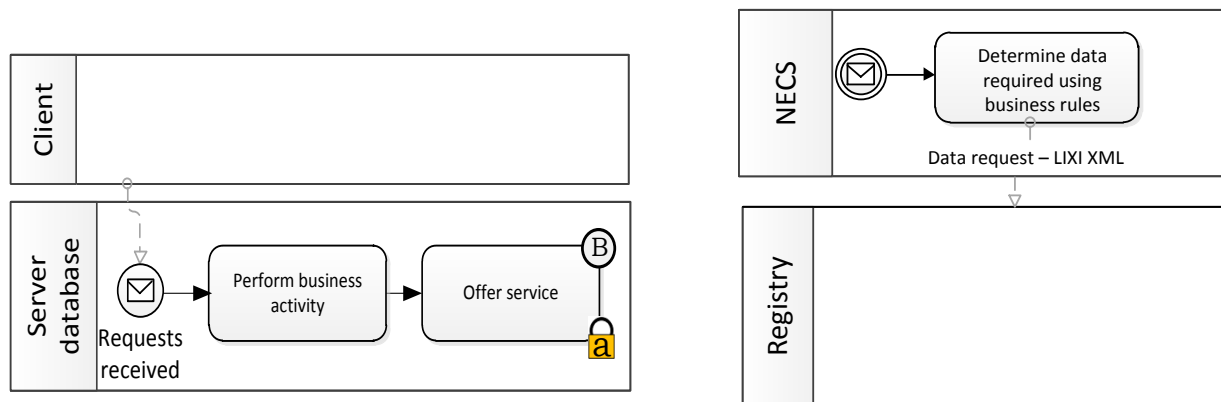
(b) 10.5 Refunds

Figure 54 Occurrence of SRP6

Occurrence of SRP7

In 6.4 Create NECS Transaction business process diagram (figure 55 (b)), the NECS receives request and *Determine data required using business rules* activity sends *Data request- LIXI XML* to Registry and waits for its fulfilment before proceeding to the next activity. In the meantime, if another request arrives at the system, then the second request is put on hold. If the reply of *Data request- LIXI XML* requires long time, then this could create a deadlock condition in the system.

The example business process diagram of SRP7 (figure 55 (a)) matches with 6.4 Create NECS Transaction (figure 55 (b)), where *Perform business activity* could be considered similar to *Determine data required using business rules* in figure 55 (b) .



(a) Example business process of SRP7

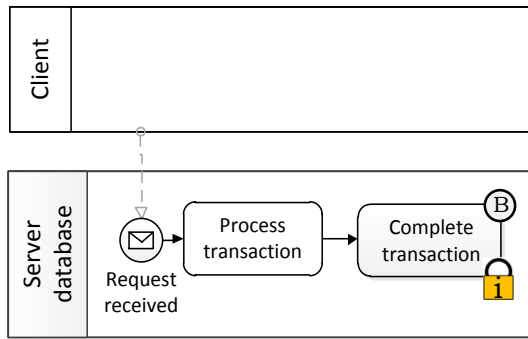
(b) 6.4 Create NECS Transaction

Figure 55 Occurrence of SRP7

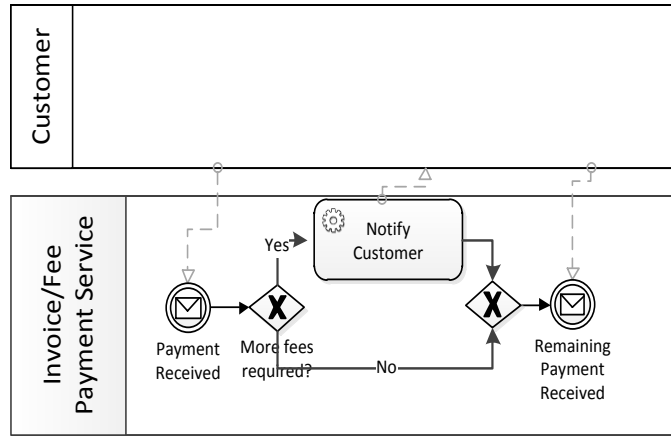
Occurrence of SRP8

In 10.3.1 Process Payment business process diagram (figure 56 (b)), the Invoice/Fee Payment Service receives payment from Customer. The service checks whether the more payment is required or not. If more payment is required, then it automatically notifies the customer. Otherwise, it proceeds to the end. In former case, if the system crashes because of receiving malicious input or due to any other reason, then the automatic notification might not be sent to the customer. As a result that particular transaction will fail, because the customer will not send the remaining payment since he might not have received the notification from the system. A failed transaction could end up writing conflicting data into database.

In SRP8 example business process diagram (figure 56 (a)), *Process transaction* activity corresponds to *More fees required* gateway and *Notifying Customer* activity present in figure 56 (b). This can be a rationale of considering figure 56 (b) as an occurrence of SRP8.



(a) Example business process of SRP8



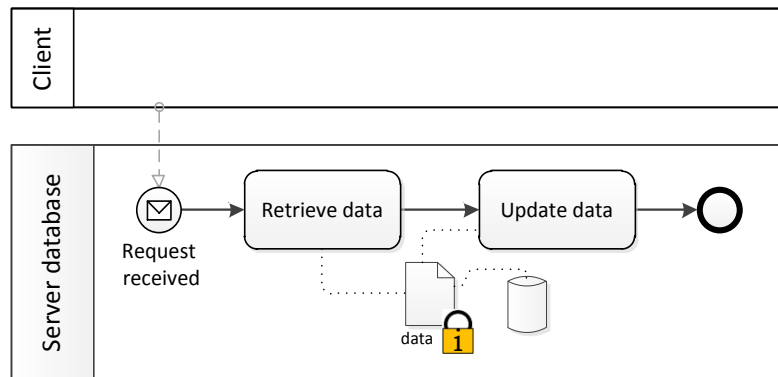
(b) 10.3.1 Process Payment

Figure 56 Occurrence of SRP8

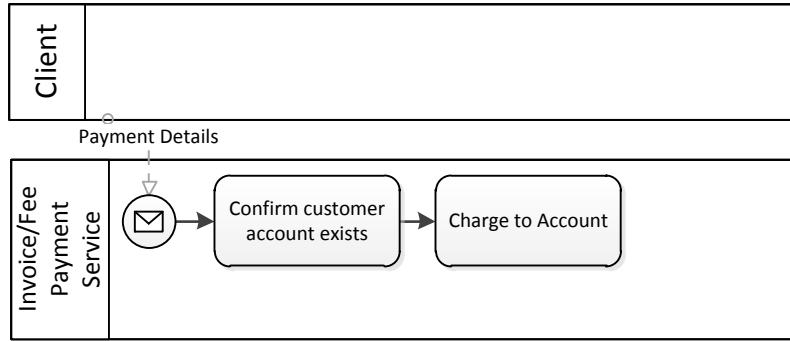
Occurrence of SRP9

In figure 57 (b), the Invoice/Fee Payment Service receives outside client request. It checks the existence of customer’s account in the system (i.e. checks the balance of the customer account), then charge the debit from it. It could be possible that the system receives multiple outside requests from the users simultaneously and executes the same debit deduction operation. If there is no clear protocol on resource access mechanism, different amount of debits can be deducted from the same balance - which can result in miscalculation.

The *Retrieve data* activity present in SRP9 example business process diagram (figure 57 (a)) corresponds to *Confirm customer account exists* activity and *Update data* could be thought similar to *Charge to Account* activity in the Process Payment business process model (figure 57 (b)).



(a) Example business process of SRP9



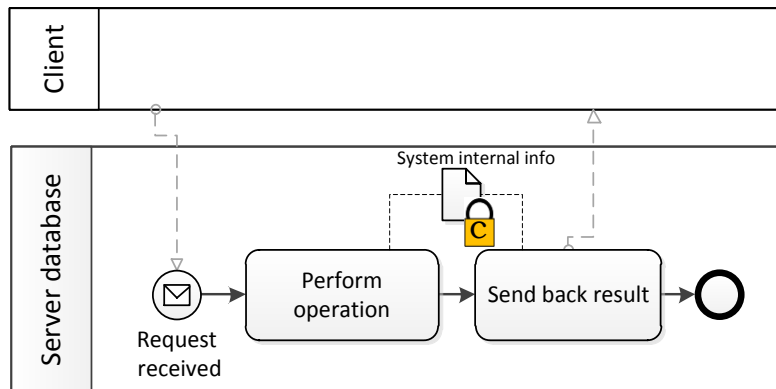
(b) 10.3.1 Process Payment

Figure 57 Occurrence of SRP9

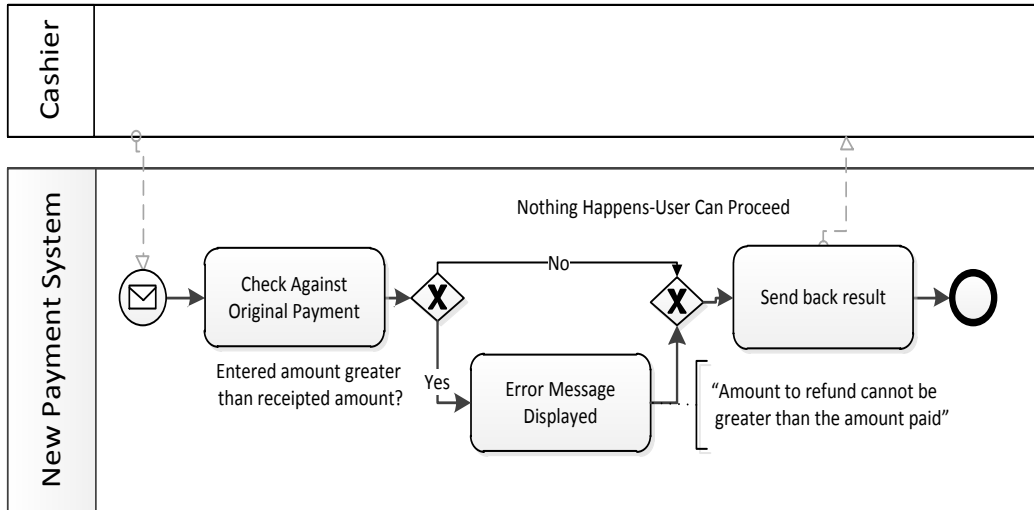
Occurrence of SRP10

In figure 58 (b), the New Payment System receives amount of payment from Cashier. It checks the amount with the original payment. If the entered amount is not greater than receipted amount, the new payment system executes in normal way and sends back the result to the user. On the other hand, if the entered amount is bigger than the receipted amount, the new payment systems displays error message. At this stage of execution, the system might leak information related to internal system to the user. The user can get detail knowledge about the system and devise more sophisticated attack techniques in order to cause harm to the system.

In figure 58, we try to elicit the similarity between the example business process of SRP10 (figure 58 (a)) and the 10.5.2 Enter Refund Amount (figure 58 (b)) business process model. Both of these processes start on receipt of outside request, perform normal operations, and finally, send back the result to the system user. These two processes could be susceptible in leaking internal system information to an attacker.



(a) Example business process of SRP10



(b) 10.5.2 Enter Refund Amount

Figure 58 Occurrence of SRP10