Raimundas Matulevičius

Fundamentals of Secure System Modelling

May 17, 2017

Springer
Chapter 11
Secure System Development Using Patterns

Security engineering requires security-related knowledge and is a time-consuming activity. However, typically, although business analysts are experts in their analysed domain, they have limited knowledge and expertise in developing secure business processes, and thus secure systems. They need to rely on best security practices, information security standards, or security experts. In this chapter we will discuss the use of security patterns to secure business processes and systems. The idea is that the majority of systems does not require new solutions; thus it is possible to reuse existing knowledge. In this way, analysts are able to introduce relevant and proper security requirements based on their rationale and later reason about security countermeasures.

This chapter, firstly, gives an overview of security pattern classifications and introduces some security risk-oriented patterns to secure business processes. Secondly, it presents a method for security requirements elicitation from business processes (SREBP).

11.1 Security Patterns

The attention to software patterns was drawn after the Gamma et al. book on design patterns [74]. Nowadays, patterns are developed for different stages and domains of software development and maintenance (e.g., software architecture, programming levels, business process management and workflow, server components, and many others). Patterns describe both the process and the phenomenon that creates this process. For instance, patterns could be used to characterise the configuration element both for software design and software architecture. Pattern presents a high-level, proven solution that resolves the given problem optimally. At the same time, patterns are generic, in some cases independent of and in other dependent on a particular implementation technology. Patterns support understanding of a problem and its solution. They could also tell a story and initiate a dialog to explain the considered problem and to influence the problem solution.
Security pattern “describes a particular recurring security problem that arises in a specific context, 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” [183] [64]. The context describes the situation, its general environment and conditions, under which the problem occurs. In the context of security, a problem occurs whenever an asset is protected in an insufficient way against abuse, or a situation arises that can allow security violations. Appropriate solutions are determined by the context, the problem and the forces of the pattern. A discussion of the benefits and drawbacks of a solution (i.e., the trade-off analysis) helps our understanding of how the forces have been resolved.

The above definition is supported by the terms of the ISSRM domain model. For instance, both system assets and business assets could be jointly used to describe the specific security context, where the security problem, expressed using the risk-related concepts, could be identified. The security solution is then defined using the risk treatment concepts.

The application of security patterns has a number of benefits [183]. Firstly, security patterns codify basic security knowledge in a structured and understandable way. Their representation is familiar to business analysts, systems engineers, security analysts and software developers, who constitute the key audience in systems development. Since patterns are already used to capture organisation and system engineering knowledge, using patterns to capture security knowledge helps us improve the integration of security into systems, where it clearly needed. Finally, using the patterns at nearly all system levels allows one to focus in a single common structure and terminology, which helps us integrate system components at different levels.

11.2 Security Pattern Taxonomy

In [183], Schumacher et al. classify security patterns into (i) enterprise security and risk management patterns, (ii) identification and authentication patterns, (iii) access control model patterns, (iv) system access control architecture patterns, (v) operating system access control patterns, (vi) accounting patterns, (vii) firewall architecture patterns, (viii) secure Internet application patterns, and (ix) cryptographic key management patterns. This taxonomy is illustrated in Fig. 11.1 and discussed below.

The scope of enterprise security and risk management patterns includea policies, directives, or constraints that apply across the enterprise. These patterns are applied for enterprise asset valuation, threat and vulnerability assessment, risk determination, enterprise security approaches and services, and enterprise partner communication.

The purpose of identification and authentication patterns is to identify an individual and confirm the individual’s identity. These patterns introduce requirements and produce design options for the identification and authentication services. Exam-
Fig. 11.1 Security pattern taxonomy, adapted from [183]

Patterns are patterns on procedural identification and authentication, physical identification and authentication, automated identification and authentication design alternatives, which also include password design and use, biometrics design alternatives, hardware token design alternatives and others.

Patterns on access control models describe security policies and security restrictions at the architectural level and the application level. The examples of these patterns are models for authorisation, role-based access control, role-right definition, reference monitoring, and multi-level security.

System access control architecture patterns handle the architecture of software systems to be secured, based on a generic set of access control requirements. They tend to introduce security solutions for separate system components or individuals. The examples include patterns on access control requirements, single access points, check points, security sessions, limited access and others.

Operating system access control patterns concern access of files. Also, they provide solutions on how operating systems should control and authorise processes when creating, monitoring, and executing system processes.

Accounting patterns deal with monitoring, registering and presenting harmful security issues that happen through operational activities. They support this with information so that appropriate solutions should be found. Examples of accounting patterns are about security accounting (and auditing) requirements and design, non-repudiation requirements and design, intrusion detection requirements and design, audit trails and logging requirements and design and others.

Firewall architecture patterns represent trade-offs between complexity, speed, and security solutions. Some may also introduce design solutions to prevent attacks
on particular network layers. Examples are patterns on proxy-based firewalls, packet filter firewalls, stateful firewalls and others.

Secure Internet applications patterns concentrate on secure Internet applications and provide guidelines for their implementation. Examples are patterns of information obscurity, protection reverse proxies, secure channels, demilitarised zones, integration reverse proxies, known partners, front doors and others.

Cryptographic key management patterns describe how to secure communication, e.g., using Internet and other distributed public services. The patterns describe cryptographic key generation, sessions and public key exchange principles.

### 11.3 Security Risk-Oriented Patterns

Security risk-oriented patterns [5] could be seen as the class of the enterprise security and risk management patterns. Since they are developed following the principles of the ISSRM domain model (see, Section 2), they support identification and valuation of system assets, determination of their security criteria, assessment of security threats, vulnerabilities, and overall risk. These patterns also support definition of security requirements and introduction of security countermeasures. Patterns are represented using the security risk-oriented BPMN (see, Section 5). Five security risk-oriented patterns are introduced in this section:

- **SRP1** Securing data from unauthorised access;
- **SRP2** Securing data that flow between business entities;
- **SRP3** Securing business activity after data is submitted;
- **SRP4** Securing business service against DoS attacks;
- **SRP5** Securing data stored/retrieved from the data store.

**SRP1: Securing data from unauthorised access.** The SRP1 pattern describes how to secure (confidential) data from access by unauthorised people or devices. In Fig. 11.2, a user requests data (i.e., a confidential business asset). In response to this request the data are retrieved (using the retrieval interface characterised as the IS asset) and provided to the user.

The problem arises (see Fig. 11.3) if retrieval of the confidential data is allowed to any user (independently of whether she or he is malicious or not) without checking his or her access permissions to the data. Such a risk event would lead to the disclosure of the confidential data: these data might be sent to business competitors, compromising the business itself. On the technical level it would bring into compromise the reliability of the data retrieval (and potentially storage) device.

To reduce this risk, the check for the access right(s) should be implemented, as illustrated in Fig. 11.4. This means that one needs to define clearance or trust levels (for accessing people or devices) and data sensitivity levels. Additionally, the verification procedure of clearance levels against sensitivity levels should be established. The implementation of these requirements would lead to application of access control (see, Section 10) models.
SRP2: Securing data that flows between business entities. A pattern addresses the electronic transmission of data between two entities, as illustrated in Fig. 11.5. Its scenario indicates that the client fills in the form and submits data through the Input interface to the Server for data employment. Here the confidentiality and integrity of data are two important security criteria; data correspond to business assets; and system assets are defined as Input interface and Server.

The assumption is made that the data are transmitted using Transmission medium (i.e., another system asset), as illustrated in Fig. 11.6. However, this situation faces (at least) two vulnerabilities. Firstly, such a transmission medium could be intercepted by an Attacker (i.e., threat agent) who has the means to intercept it by acting as a proxy. Secondly, since data are not encrypted, they could be misused, for example, modified and passed to the Server. This event would harm data, would lead to the loss of transmission medium reliability, and would negate data integrity (and confidentiality).
A potential risk treatment decision includes risk reduction by making data unreadable and verifying the received data. In Fig. 11.7, these security requirements are introduced to reduce the identified risks. The implementation includes introduction and application of the cryptographic algorithms and the checksum algorithms.

Another risk treatment decision, risk avoidance, could be applied, resulting in the change of the transmission medium. This could result in, for example, physical data delivery.

**SRP3: Securing business activity after data is submitted.** This ensures valid data entry into business processes by rejecting the unwanted malicious data. As illustrated in Fig. 11.8, it secures the business activity (i.e., any activity after data is submitted) of which integrity and availability have to be ensured. Hence, the data could be submitted by any entity to the input interface. The activity Submit data is vulnerable, as shown in Fig. 11.9, because it does not check the incoming data, which could be submitted by an **Attacker** (i.e., threat agent) who is capable of writing malicious scripts (e.g., cross-site scripting, like SQL queries or XPath injections)
Fig. 11.6 SRP2: risk modelling (threat to integrity)

![Diagram](https://example.com/diagram1.png)

Fig. 11.7 SRP2: risk treatment modelling

![Diagram](https://example.com/diagram2.png)

[75]). Executing these scripts enable an attacker to read/write the confidential business data or change the business rules. If this happens, it risks the confidentiality and integrity of the data itself, and any activity after data is submitted may be harmed, become unavailable or lose its integrity; additionally the input interface would be compromised.

To mitigate the risk(s), the pattern introduces a security requirement of filtering the incoming data, as illustrated in Fig. 11.10. This security requirement could be implemented in input validation [75], input sanitisation [72], input filtration [75], or/and input canonicalization [42] security controls.

**SRP4: Securing a business service against DoS attacks.** The pattern ensures the availability of a business service by protecting the IS from a denial of service (DoS) attack [122]. The major idea is to protect the business services (i.e., business assets), which are provided by a server, in order to guarantee the availability of this business service, as illustrated in Fig. 11.11. The pattern assumes that there exists an attacker capable of hacking a large number of computers that simultaneously
request this business service (see Fig. 11.12). The attacker is able to target the server because the used protocol (e.g., TCP, ICMP, or DNS) allows handling unlimited number of requests for a service [37]. Hence, the impact of the risk event is that the server becomes incapable of operating resulting in the business service becoming unavailable to ordinary users.

To reduce the DoS attack, one needs to implement a security requirement for checking for abnormal requests, as shown in Fig. 11.13. This requirement would include filtering and classifying of incoming requests, detecting abnormal requests, and discarding the attacking ones. Implementation of the requirements would result in detection [157] [105], filtering [202], and response to attack [124] techniques.

**SRP5: Securing data stored in/retrieved from the data store.** The pattern ensures the data privacy at the data store from insiders (i.e., administrators or malware that could infect the data store). The main goal of this pattern is to prevent the flow control of information from leaking horizontally across the departments (i.e., at the same level). Let’s assume (see Fig. 11.14) that there exists a storing/retrieval interface (i.e., system asset) which helps clients (i) to store the client’s data (i.e., business asset) in the data store and (ii) to retrieve them when needed.
Let’s assume that there exists an attacker, i.e., a malicious insider who has privileges to access the data store and also retrieve data directly from it (see Fig. 11.15). If the storing/retrieval interface (including the queries to the database) are designed in a way that data are stored/retrieved in a plain format, the insider could view the client’s data, thus negating the data confidentiality.

To reduce this security risk, one needs to introduce security requirements (see Fig. 11.16) that help in making data invisible before they are stored in the data store and making them visible after they are retrieved from the database. A security requirement for monitoring for malicious changes at the data store should also be introduced. Implementation of these security requirements would result in application of auditing, cryptographic, and/or data protection techniques [87].
11.4 Security Requirements Elicitation from Business Processes

A method for security requirements elicitation from business processes SREBP suggests means to derive security requirements from the business processes by applying security risk-oriented patterns [1]. In this section we will present the major principles of the SREBP method and will illustrate it through the Football Federation case.
11.4 Security Requirements Elicitation from Business Processes

11.4.1 SREBP Method

To present the SREBP method [1][179] we will apply the component-based method view [79], which requires the description of method perspective (procedure for the modelling process from a particular perspective), framework (relationships between the individual method components), cooperation principles (a range of specialist
skills for cooperation between different roles) and all method components (consisting of concepts, procedures, and notations).

**Perspective.** The major goal of the SREBP method is to identify the organisation’s assets, determine its security objectives, and elicit its security requirements. Using the method, the business/security analysis can derive security requirements from the business processes and introduce them as security restrictions back to the business processes.

**Cooperation among stakeholders.** Typically, security engineering requires a close collaboration between the business analyst (i.e., the specialist of the business domain) and the security analyst (i.e., the specialist of the security domain) [5] [179]. Being experts in the business domain, business analysts have limited or no expertise in security engineering. They have to rely on best security practices, information security standards, or security experts.

The business analyst introduces to the security analyst the business context and describes the enterprise’s workflow. In this way the security analyst (in collaboration with the business analyst) identifies what the business assets are, what security objectives (in terms of confidentiality, integrity, and availability) should be taken into account, and what the system assets to support the identified business assets are.

Once the security requirements are derived from the business process models, they can be used to annotate the original business process models. The annotated business model is returned back to the business analyst. But the feedback could also include security risk models. Another collaboration might be on security requirements trade-off analysis. However, this activity is not emphasised in the SREBP method.

**SREBP components.** The SREBP method components are listed in Table 11.1. The majority of these concepts, i.e., business asset, security criterion, system asset, security risk, and security requirements, are taken from domain model for the information systems security risk management (see, Section 2). But the SREBP
method also includes a few concepts, i.e., *security risk-oriented patterns*, *pattern occurrence*, *security model*, which result from the application and performance of the base ISSRM concepts.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Procedures</th>
<th>Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value chain</td>
<td>Created by the business analyst, expresses how the enterprise business functions are related in order achieve enterprise’s goals</td>
<td>BPMN</td>
</tr>
<tr>
<td>Business process diagram</td>
<td>Created by the business analyst, expresses the use of the computerised information system. These diagrams should express the use of data objects, data flows and data stores</td>
<td>BPMN</td>
</tr>
<tr>
<td>Business asset</td>
<td>Identified from the value chain</td>
<td>Textually and/or graphically</td>
</tr>
<tr>
<td>Security criterion</td>
<td>Identified by understanding importance of the business assets</td>
<td>Textually and/or graphically</td>
</tr>
<tr>
<td>System asset</td>
<td>Identified when analysing the business process diagrams</td>
<td>Textually and/or graphically</td>
</tr>
<tr>
<td>Security risk (and its major components)</td>
<td>Identified from the business process diagrams by instantiating the security risk-oriented patterns</td>
<td>Security risk-oriented BPMN</td>
</tr>
<tr>
<td>Security requirements</td>
<td>Identified from business process diagrams by applying the security risk-oriented patterns and by instantiating pattern security parts</td>
<td>Documented textually as security requirements statements, and graphically using UML notations depending on the analysed contextual area</td>
</tr>
<tr>
<td>Security risk-oriented patterns</td>
<td>Artifact used to guide security risk requirements derivation from the business process diagrams. The patterns describe recurring security risks that arise within business processes. To mitigate the risks, the patterns recommend security requirements</td>
<td>Documented textually in the structured template [5] and graphically using security risk-oriented BPMN (see Section 11.3)</td>
</tr>
<tr>
<td>Pattern occurrence</td>
<td>Identified in the business process diagram using security risk-oriented patterns</td>
<td>Highlighted in the analysed business process diagram</td>
</tr>
<tr>
<td>Security model</td>
<td>Derived from the business process model and the result of security risk-oriented pattern application</td>
<td>Represented graphically using UML notations depending on the analysed contextual area and applied pattern</td>
</tr>
</tbody>
</table>

In the second column of Table 11.1, the procedures used to identify the relevant concepts are presented. Hence the business analyst creates *value chain* and *business process diagrams* as the part of the organisation’s business process management. The *asset-related concepts* are identified from the *value chain* and *business process diagrams*, and *security risk-related and risk treatment-related concepts* are defined using the *security risk-oriented patterns*.

The third column presents the notations used to represent concepts. A notable set of concepts is expressed using textual language, which is supported with tar-
geted graphical notations. Since SREBP is meant to consider business processes, the majority of the notations are BPMN or security risk-oriented BPMN. The security requirements models are represented using UML.

**SREBP conceptual framework.** In Fig. 11.17, the relationships between the SREBP method components are described. Hence, the *Business process diagram* expands the *Value chain diagram*. The *Business assets* are elicited from the *Value chain*. The security analyst in cooperation with the business analyst determines the *Security objective* for each identified *Business asset*. *System asset* supports *Business asset*, which are also refined when considering the *Business process diagram*. When applying Security risk-oriented patterns, *Pattern occurrences* are found in the *Business process diagram*. *Pattern occurrences* result in a *Security model*, which is extracted from the *Business process diagram* based on the used Security risk-oriented pattern. *Security requirements* are derived from the *Security model* and they define the security constraints on *Assets*.

**Fig. 11.17** The SREBP framework, adapted from [179]

Figure 11.18 presents an SREBP process. It consists of two stages: *(i)* business asset identification and security objective determination, and *(ii)* security requirements elicitation. The second stage includes three activities: identify patterns, extract security model and derive security requirements.

**Fig. 11.18** The SREBP process, adapted from [179]
During pattern identification one need to find pattern occurrences in the business process diagram. One could apply methods of hierarchical level matching, business perspective matching, structural similarity and semantic similarity matching [4]. Once the pattern occurrences are determined, one could extract the security model. Depending on the chosen security risk-oriented pattern, different activities for security model extraction could be performed (see, Fig. 11.19). After extracting the security model, it becomes possible to derive and document security requirements.

![Fig. 11.19 SREBP activity Extract security model, adapted from [179]](image)

### 11.4.2 Pattern Application

In this section, we will analyse the running example – the extract of the Football Federation case presented in Chapter 1. We will apply the SREBP method and illustrate how security risk-oriented patterns could help in determining security requirements.

To start performing the security requirements elicitation one needs to collect knowledge of an organisation’s values from the value chain and the business functions. The value chain is presented in Fig. 1.6. In Fig. 11.20 a detailed workflow of Register game report process is given. The process has two business partners (Umpire and FootballFederationEmployee) expressed as swimlanes, while ERIS is identified as a system.

#### 11.4.2.1 Business Asset Identification and Security Objective Determination

The first stage starts with the analysis of the value chain (see Fig. 1.6) from which the assets that must be protected against security risks are determined. The stage requires collaboration between security analysts and the stakeholders from the analysed enterprise. It consists of two activities:

i) **Identify business assets:** During this activity the central artefacts considered in the value chain are identified. Typically, further details of these artefacts are consid-
ereed in the business process model. The value chain can either have a single artefact used in all the processes or be comprised of multiple artefacts in each operational business process. In the Football Federation case, it is possible to identify important artefacts from each business activity. The protected assets include Team, Player, Umpire, League and region, Timetable, and Game. In this discussion we will focus on Game (and its registration process).

**ii) Determine security objectives:** The activity addresses the determining of key security objectives – confidentiality, integrity and availability – for identified business assets. The following security objectives for business asset Game are defined:

- *i) Game* should be confidential (at least some its attributes or at least at some stage in its existence), i.e., no unauthorised individual should read it and its relevant data;
- *ii) Game* should be integral, i.e., the Game and its relevant data should not be tampered; and
- *iii) Game* and its relevant data should be available to the business partners at any time.
11.4.2.2 Security Requirements Elicitation

At the second stage, the security requirements elicitation is performed using security risk-oriented patterns. It is important to note that each artefact—data or process—separately considered and protected by the patterns, contributes to the security of the business asset (i.e., Game) identified at the first stage.

Securing data from unauthorised access. The major concern of this pattern is to protect the confidentiality of the identified business asset, in our example the Game, when it is being manipulated by the system asset (i.e., the ERIS). The security threat arises if the Game and its attributes (like (Game info, Gamereport, and Confirmation) are accessed by users who does not have access permissions. The risk event would: (i) negate confidentiality of Game, (ii) lead to unintended use of the Game data, and (iii) harm the ERIS’s reliability.

A way to mitigate the security risk is through the introduction of an access control mechanism, for example, the Role-Based Access Control (RBAC) model (see, Chapter 10). The RBAC model (see, Fig. 11.21) is elicited by performing the following activities:

Fig. 11.21 RBAC security model - register game report business process

i) Identify resource: Hence, the business asset (i.e., Game) is defined as a resource that needs to be protected from unauthorised access. The protected resource is characterised by its attributes—gameInfo, gameReport and confirmation—that add value to the asset.
ii) Identify role: The swimlanes are considered as an outside role while the lanes of an information system corresponds to an internal role. We consider both outside and internal roles, since they both could access the secured business asset i.e., Game. These roles (e.g., Umpire and FootballFederationEmployee) are modelled using the «role» stereotype in RBAC security model (see Fig. 11.21).

iii) Assign users: This activity assigns roles to users, which are instances of some role. Usually it is not possible to elicit concrete users from the operational business process. This requires expertise of and collaboration with domain experts.

iv) Identify secured operation: An operation is an executable set of actions that can change the state of the protected resource. For instance, createGame, updateGameReport and updateConfirmation are secured operations.

v) Assign permissions: Permissions specify the security actions – namely, Insert and Update – over secured operations that the role can perform to change the state of the protected resource. For example, the Umpire role has permission to update Game’s attribute gameReport.

The developed security model (see Fig. 11.21) suggests the refinement of the security requirement check for the access rights to the following context-specific security requirements:

- **SecReq.1:** Umpire should be able to update the gameReport.
- **SecReq.2:** FootballFederationEmployee should be able to insert the Game (i.e., create a new instance of Game, including gameInfo, gameReport, and confirmation).
- **SecReq.3:** FootballFederationEmployee should be able to update the confirmation.

The security model (i.e., Fig. 11.21) defines how authorised parties should access the protected resources. However, it does not support capturing scenarios like entailment constraints [91], delegation constraints [19] and usage control [165]. These requirements could be determined in the collaboration between business and security analysts.

Securing data that flows between business entities is used to exchange data between business partners (e.g., Umpire and Football Federation Employee) and the system (e.g., ERIS). Here, data, like gameInfo, different notifications, gameReport, confirmation, etc., need to be protected when they are transmitted over the (untrusted) communication channel, i.e., Internet. To define security requirements, one needs to perform two activities:

i) Identify communicators: Communicators are the entities that transmit or receive data. Operational business processes are considered to identify the system and their business partners who exist outside of an organisation but transmit/receive data to/from the organisation. In Fig. 11.22, we illustrate a security model for communication channel between ERIS and Umpire using a UML interaction diagram. ERIS is modelled as the system that communicates with the Umpire identified as the business partner.

ii) Identify data transmission: One needs to determine the business asset and/or its relevant data transmitted or received between the identified communicators over
the untrusted communication channels, i.e., Internet. For example, *game report* is communicated between Umpire and ERIS; thus, they require to be protected.

The above activities result in the following security requirements for the Umpire and ERIS and correspondingly for other entities (e.g., Football Federation Employee, Team Representative, etc.) that communicate with ERIS:

- **SecReq.4**: ERIS should have unique identity in the form of key pairs (public key, private key) certified by a certification authority.
- **SecReq.5**: Umpire should encrypt and sign *game report* (and other data communicated to ERIS) using keys before sending it to ERIS.

![Fig. 11.22 TLS protocol implementation, adapted from [16]](image)

A security requirements implementation could be fulfilled by the standard transport layer security (a.k.a., TLS) protocol [16] (see Fig. 11.22). As the first contact, the Umpire sends ERIS a *handshake* message, which includes a random number. Then the ERIS responds with its public key and information about the certification authority. After verification of the ERIS’s public key, the Umpire generates the secret and sends it to the ERIS encrypted with the ERIS’s public key. The ERIS then decrypts the secret using the private key and generates symmetric session keys. The keys enable Umpire and ERIS to establish a secure session for data exchange. The encryption keeps the transmitted data (e.g., *game report*, etc.) confidential, and signing it ensures that the received data is not tampered. The secure communication continues until it is not explicitly terminated by Umpire or ERIS.

**Securing business activity after data is submitted** ensures that the input data submitted by business partners are correct and complete. In this contextual area two activities are suggested:

1. **Identify input interfaces**: The activity identifies the system input interfaces from the operational business processes that have incoming message flows. The input interfaces are those activities of the information system that receives input from the enterprise stakeholders.
ii) Identify input data: The activity identifies the input data received by the input interfaces from the enterprise’s business partners.

In the Football Federation case, Update game report could be treated as the input interface of ERIS that receives the game report from Umpire. The threat agent can exploit the vulnerability of the input interfaces by submitting the data with malicious scripts. If this happens the availability and integrity of any activity (e.g., Update confirmation) after the input interface (e.g., Update game report) may be misused or negated. To avoid this risk the following security requirements are suggested:

- **SecReq.7**: Update game report should filter the input (i.e., game report).
- **SecReq.8**: Update game report should sanitise the input (i.e., game report) to transform it to the required format.
- **SecReq.9**: Update game details should canonicalise the input (i.e., game report) to verify against its canonical representation.

Input filtration (i.e., SecReq.7) validates the input data against the secure and correct syntax [42]. The string input should be checked for length and character set validity (e.g., allowed and blacklisted characters). The numerical input should be validated against their upper and lower value boundaries. Input sanitisation (i.e., SecReq.8) should check for common encoding methods used (e.g., HTML entity encoding, URL encoding, etc.). The input canonicalisation (i.e., SecReq.9) verifies the input against its canonical representation [42].

**Securing business service against Dos attacks** secures the network infrastructure of the system. The system is composed of several small functional units, which can be deployed at either a single location or multiple locations connected through the Internet. The goal is to guarantee availability of these functional units. Two activities are performed:

i) Identify functional-unit: A functional unit is an activity or sub-process implemented on an independent network infrastructure to provide certain functionality to an enterprise’s information system. A system can be comprised of one or more functional units. For instance, the ERIS system (see Fig. 11.20) consists of three functional-units, Create game, Update game report, and Update confirmation, which could be deployed on an independent network infrastructure connected through the Internet to form a single system (i.e., ERIS). Here, the elicitation of security requirements for Update game report is illustrated (see Fig. 11.23).

ii) Identify business partner: Business partners are the external entities that can access the network infrastructure in order to communicate with the enterprise information system. The access involves any request type necessary to receive or send data. For instance, Umpire could be treated as an external entity that communicates with ERIS.

In Fig. 11.20, ERIS has a functional unit Update game report offered to Umpire. The threat agent may exploit the hosts in the channel and hack them because of the protocol (e.g., TCP, ICMP or DNS [37]) vulnerability, i.e., the ability to handle an unlimited number of requests for service. When receiving simultaneously multiple requests, ERIS will not be able to handle them; thus, the services become
unavailable. The above activities help us develop a security model (see Fig. 11.23) that defines three types of firewalls [183] – Packet Filter Firewall, Proxy Based Firewall and Stateful Firewall. The security model introduces the following requirements to mitigate the risks:

- **SecReq.10:** Update game report should establish a rule base (i.e., a collection of constraints used by different firewalls) to communicate with the Umpire.
- **SecReq.11:** Packet Filter Firewall should filter the Umpire’s address to determine if that is not a host used by the threat agent.
- **SecReq.12:** Proxy Based Firewall should communicate to the proxy which represents Update game report to determine the validity of the request received from Umpire.
- **SecReq.13:** Stateful Firewall should maintain the state table to check the Umpire’s request for additional conditions on established communication.

It is important to notice that the communication between the Umpire (and also Football Federation Employee) and the ERIS is bidirectional. Similar requirements must be taken into account when ERIS sends messages back to the business party.

**Securing data stored in/retrieved from the data store** is used to define how data are stored in and retrieved from the associated databases (e.g., Game storage). If the threat agent is capable of accessing and retrieving the data, their confidentiality and integrity would be negated resulting in harm to the business asset (i.e., the Game report) and its supporting system assets (i.e., ERIS). To prevent unauthorised access to the datastore, the access control model could be defined:

1) **Identify Datastore resource:** For example, Game storage is identified as a single collective resource. The identified business assets and their related data in the operational process models are modelled as resource attributes. This means that the attributes Game info, Game report, and Confirmation will represent the attributes of business asset Game.
ii) Identify Datastore's operations: The activity identifies operations that save and retrieve the data identified in previous activity to and from Game storage. These operations are modelled as operations of Game storage's resource.

Once the resource and operations are modelled, the activities identify role and assign permissions are performed as described earlier (see discussion on securing data from authorised access). This results in a security RBAC model for Game storage, given in Fig. 11.24.

Once we identify access control policies, it becomes possible to introduce security requirements for monitoring and auditing (i.e., monitor for malicious changes) and for hiding data/ and making it visible (i.e., make data invisible and make data visible), as illustrated in Fig. 11.16. The following security requirements should be taken into account:

- **SecReq.14**: The ERIS should audit the operations after the retrieval, storage or any other manipulation of data in the Game storage.
- **SecReq.15**: The ERIS should perform operations to hide/unhide data when they are stored/retrieved to/from the Game storage.

Auditing (supported by the access control policy) is the process of monitoring and recording selected events and activities [154]. It determines who performed what operations on what data and when. This is useful for detecting and tracing security violations performed on gameInfo, gameReport, and confirmation.

A possible implementation of SecReq.15 is through cryptographic algorithms. The encryption offers twofold benefits: (i) the data would not be seen by the Game storage users (e.g., by the database administrator) where the circumstances do not allow one to revoke their permissions; (ii) due to any reason if someone gets physical access to the Game storage she or he would not be able to see the confidential data stored.

### 11.4.2.3 Once Security Requirements Are Elicited

Once security requirements are elicited, one needs to understand whether they all need to be implemented. This basically includes requirements prioritisation and security trade-off analysis. For instance, implementing all security requirements might be costly. In addition, although this can guarantee a certain level of security, it would influence the efficiency and performance of the system. Thus the optimal decision on security requirements needs to be found once requirements are derived by applying the security risk-oriented patterns.

### 11.5 Further Reading

A classical overview of software requirements patterns is presented in [210]. After introducing the notion of patterns for software requirements, the author describes
the pattern taxonomy for information, user functions, data, performance, flexibility and access control. The security pattern taxonomy [183] viewed over in this chapter is one possible classification. Elsewhere, [25] differentiates between available system patterns and protected systems patterns. The first group is concerned with system availability and provides security approaches that give continuous access to the services and resources. The second group concerns protection of privileged resources by providing a set of methods that protect valuable assets against unauthorised use. In [52], secure design patterns are introduced and classified into (i) architectural-level patterns to describe the high-level distribution of the tasks and communication protocols between different modules; (ii) design-level patterns to design standards of high-level system modules; and (iii) implementation-level patterns to deal with low-level security concerns. These patterns are appropriate for adjusting the application of a particular functionality in the system. To support pattern classification, Slavin et al. suggest feature diagram hierarchies to support more confident and expert-like decisions in efficient time [195].

Nine modelling approaches are compared in terms of modelling constructs to represent security patterns [20]. Elsewhere, in [22], a framework to link security standards with a security engineering method is proposed. The author illustrates how goal-oriented Si* Secure Tropos [77] [78] [126] [177] [18], CORAS [123], and problem frames [96] could be used to ensure compliance with security standards. Security risk-oriented patterns, discussed in this chapter, could be considered within different model representations. For instance, besides pattern application in business processes, application of SRPs is illustrated within the models of security risk-aware Secure Tropos [176]. The study illustrates how SRP could initiate security requirements derivation and security trade-off analysis.
11.6 Exercises

Exercise 11.1: What are the similarities and differences between the different security pattern taxonomies: STRIDE [189], CAPEC [146], security threat taxonomy for distributed systems [206], pattern taxonomy suggested by Schumacher et al. [183], etc.?

Exercise 11.2: Based on your experience, develop a new security risk-oriented pattern.

Exercise 11.3: For the Football Federation process model (see Fig. 1.5), apply the method of security requirements elicitation from business processes (SREBP) and derive the security requirements for the Team asset.

As illustrated in Fig. 1.5, Team asset is characterised by the attributes

- teamInfo – team info;
- teamRep – team representative;
- participationDecision – decision about participation; and
- regionAndLeague – region and league, where team decided to participate.

The solution should illustrate the application of security risk-oriented patterns and include:

- SRP1: Security model (i.e., RBAC model) and the derived list of security requirements
- SRP2: Security model (i.e., collaboration model) and the derived list of security requirements
- SRP3: Derived list of security requirements (describing the input filtering)
- SRP4: Security model (i.e., class diagram) and derived list of security requirements (describing the requirements for the firewall architecture)
- SRP5: Security model (i.e., RBAC model regarding the database) and derived list of security requirements (including auditing and information hiding requirements).

Exercise 11.4: Discuss how security risk-oriented patterns could be defined using:

- Security Risk-aware Secure Tropos;
- Security Risk-oriented Misuse Cases;
- Mal-activities for security risk management.
Chapter 13

References

34. BSI: BSI Standard 100-3 version 2.5. Risk Analysis Based on IT-Grundschutz. Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn (2008)
74. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (1994)
References


146. MITRE: Common Attack Pattern Enumeration and Classification URL: https://capec.mitre.org