Security requirements
What needs to be secured?

Raimundas Matulevičius

Outline

• What are requirements?
• What are security requirements?
• Security requirements modelling?
• SREBP
• Patterns and Examples
Outline

- What are requirements?
- What are security requirements?
- Security requirements modelling?
- SREBP
- Patterns and Examples

Definition of RE

Requirements Engineering (RE) is a set of activities concerned with identifying and communicating the purpose of a software-intensive system, and the contexts in which it will be used. Hence, RE acts as the bridge between the real world needs of users, customers, and other constituencies affected by a software system, and the capabilities and opportunities afforded by software-intensive technologies.

- Not a phase or stage!
- Communication is as important as the analysis
- Quality means fitness-for-purpose. Cannot say anything about quality unless you understand the purpose
- Designers need to know how and where the system will be used
- Requirements are partly about what is needed...
- ...and partly about what is possible
- Need to identify all the stakeholders - not just the customer and user
Where do we start?

- **Identify the problem**
  - what is the objective of the project?
  - the “vision” of those who are pushing for it?
    - e.g., “Meeting scheduling is too costly right now”

- **Scope the problem**
  - given the vision, how much do we tackle?
    - e.g. “Build a system that schedules meetings”, ...or...
    - e.g. “Build a system that maintains people’s calendars” ...or...

- **Identify solution scenarios**
  - given the problem, what is the appropriate business process for solving it?
    - e.g. “Anyone who wants to schedule a meeting goes to the secretary, gives details and the secretary handles the rest”, ...or...

- **Scope the solution**
  - Given a business process, what parts should be automated, and how?
    - e.g. “Computer takes in scheduling request details, outputs a solution” ...or...
    - e.g. “Solution arrived at interactively by secretary and computer” ...or...
Requirements Elicitation

- **Starting point**
  - Some notion that there is a “problem” that needs solving
    - e.g. dissatisfaction with the current state of affairs
    - e.g. a new business opportunity
    - e.g. a potential saving of cost, time, resource usage, etc.

- **Collect enough information to**:
  - **identify the “problem”/“opportunity”**
    - Which problem needs to be solved? (identify problem Boundaries)
    - Where is the problem? (understand the Context/Problem Domain)
    - Whose problem is it? (identify Stakeholders)
    - Why does it need solving? (identify the stakeholders’ Goals)
    - How does the problem manifest itself? (collect some Scenarios)
    - When does it need solving? (identify Development Constraints)
    - What might prevent us solving it? (identify Feasibility and Risk)

  - **become an expert in the problem domain**
    - Learn how to find your way round a new problem area quickly
    - Use your (initial) ignorance as an excuse to ask (dumb?) questions
    - Recognise the domain expertise of the people you talk to

Elicitation Techniques

- **Traditional techniques**
  - Reading existing documents
  - Analyzing hard data
  - Interviews
    - •Open-ended
    - •Structured
  - Surveys / Questionnaires
  - Meetings

- **Collaborative techniques**
  - Focus Groups
    - •Brainstorming
    - •JAD/RAD workshops
  - Prototyping
  - Participatory Design

- **Contextual (social) approaches**
  - Ethnographic techniques
    - •Participant Observation
  - Discourse Analysis
    - •Conversation Analysis
    - •Speech Act Analysis

- **Cognitive techniques**
  - Task analysis
  - Protocol analysis
  - Knowledge Acquisition Techniques
    - •Card Sorting
    - •Laddering
    - •Repertory Grids
    - •Proximity Scaling Techniques
What are requirements?

- **Domain Properties:**
  - things in the **application domain** that are true whether or not we ever build the proposed system

- **Requirements:**
  - things in the **application domain** that we wish to be made true by delivering the proposed system
    - Many of which will involve phenomena the machine has no access to

- **A Specification:**
  - is a description of the behaviours that the **program** must have in order to meet the requirements
    - Can only be written in terms of shared phenomena!
Outline

• What are requirements?
• What are security requirements?
• Security requirements modelling?
• SREBP
• Patterns and Examples

What is a security requirement?

• A requirement defining **what level** of security is expected from the system with respect to some type of threat or malicious attack
  • what you require?
    – Different from the **security objective** (criterion)
      • not why it is needed?
    – Different from the choice of **security controls** (design)
      • not how to achieve it?
  • vs. **related concepts:**
    – **Security**: malicious / Intended harm
    – **Safety**: accidental harm
    – **Dependability**: More general concept covering both safety, security and several other concepts (e.g., availability, reliability, robustness, survivability ...)
    – **Privacy**: sometimes considered as a subcategory of security, but sometimes also in conflict
Outline

• What are requirements?
• What are security requirements?
• Security requirements modelling?
• SREBP
• Patterns and Examples
Security Modelling Languages

- Early requirements
- Late requirements
- Architectural design
- Detailed design
- Implementation and testing

- BPMN
- Secure TROPOS
- KAOS extension to security
- Misuse cases
- Mal-activity diagrams
- UMLsec
- SecureUML

Business Process Modelling

- **Approach**
  - What organisation needs to do to achieve their business objectives?

- **Advantages**
  - Reasonably intuitive
  - Explicit declaration of business activities, processes and sub-processes

- **Disadvantages**
  - Captures only a dynamic picture
  - Not focussed on the business support by technology
Business Process Model and Notation
version 2.0

- Descriptive Modelling
- Analytical Modelling
- Executable Modelling

Security risk management process
Context and Assets Identification

- **Description of organisation and its environment**
  - sensitive activities related to information security

Security Objectives Determination

- **Determine the security objectives to be reached**
  - Confidentiality, Integrity, Availability
Risk Analysis and Assessment

- Identify risks and estimate them qualitatively or quantitatively

Risk Treatment Decisions

<table>
<thead>
<tr>
<th>Risk Treatment decisions</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoiding risk</td>
<td>Decision not to be involved in, or to withdraw from a risk</td>
</tr>
<tr>
<td>Transferring risk</td>
<td>Sharing with another party the burden of loss for a risk</td>
</tr>
<tr>
<td>Retaining risk</td>
<td>Accepting the burden of loss from a risk</td>
</tr>
<tr>
<td>Reducing risk</td>
<td>Action to lessen the probability, negative consequences, or both, associated with a risk</td>
</tr>
</tbody>
</table>
Security Requirements Definition

- Security requirements - security solutions to mitigate the risks
- If security requirements are unsatisfactory
  - Revise the risk treatment step
  - Revise all of the preceding steps

Outline

- What are requirements?
- What are security requirements?
- Security requirements modelling?
- SREBP
- Patterns and Examples
Understanding work practices and their changes

**Processing of Information**

Everything that IT does, reduces to **six** functions

- **Capturing information**
  - Keyboard, bar code reader, digital camera

- **Transmitting information**
  - Wired-, wireless-phone

- **Storing information**
  - Hard disk, memory card, internet

- **Retrieving information**
  - From any storage device

- **Manipulating information**
  - Calculations, combinations of data

- **Displaying information**
  - Monitor, printer

**SREBP**: Security Requirements Elicitation from Business Processes
Outline

• What are requirements?
• What are security requirements?
• Security requirements modelling?
• SREBP

Patterns and Examples

Security Patterns

• Arises in a specific security context
• Describes a particular recurring security problem
• Presents a well-proven generic scheme for a security solution
SRP1: Securing data from unauthorised access

Example
Example

Registry

Process Selected Business Process(es)

Plan Number

Pre allocate Plan Numbers

Digital Data

Extract Digital Data

Auto Validate Plan

Validate Plan

Plan Number

Datastore

Fee Calculation Service

Send Digital Data

Plan Number Allocated

Validate Address Service

Send Validation Report

Corporate Databases

Schema, Business rules

Topographical, Survey

Tenure Imagery & Title

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

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

RQ3. Secured operations (e.g., *Fee Calculation Service*) should be performed by different users assigned to the same role.

RQ4. A sequence of secured operations (e.g., *Pre allocate Plan Numbers and Send Plan Number*) should be performed by the same user assigned to the role (e.g., *Planning Portal*).

RQ5. The system (i.e., *Registry*) should place constraints on how confidential data should be used by the roles (i.e., *Lodging Party* and *Planning Portal*).

SRP2: Securing data that flow between the business entities

- **Asset**
SRP2: Securing data that flow between the business entities

- Risk

SRP2: Securing data that flow between the business entities

- Risk-treatment
Example
**Example**

**RQ6.** The server (e.g., *Registry*) should have the unique identity in the form of key pairs (public key, private key) certified by a certification authority.

**RQ7.** The client (e.g., *Lodging Party* and *Planning Portal*) should encrypt and sign the data (e.g., *Selected Process(es), Plan Number, and other*) using keys before sending it to the server (e.g., *Registry*).

**SRP3:** Securing business activity after data is submitted
Example

RQ8. The input interface (e.g., Fee Calculation Service) should filter the input data (e.g., Payment Consent).

RQ9. The input interface (e.g., Fee Calculation Service) should sanitize the input data (e.g., Payment Consent) to transform it to the required format.

RQ10. The input interface (e.g., Fee Calculation Service) should canonicalize the input data (e.g., Payment Consent) to verify against its canonical representation.

SRP4: Securing business activity after data is submitted
RQ11. Server (e.g., Registry) should establish a rule base (i.e., a collection of enterprise' constraints used by different firewalls) to communicate with the business partners (e.g., Planning Portal).

RQ12. Packet Filter Firewall should filter the business party’s (e.g., Planning Portal’s) address to determine if it is not a host used by the threat agent.

RQ13. Proxy Based Firewall should communicate to the proxy which represents the business service (e.g., Pre allocate Plan Number) to determine the validity of the request received from the business party (e.g., Planning Portal).

RQ14. State Firewall should maintain the state table to check the party’s (e.g., Planning Portal’s) request for additional conditions of established communication.
SRP5: Securing data stored in/retrieved from the data store

Example

**RQ15.** The server (e.g., Registry) should audit the operations after the retrieval, storage or any other manipulation of data in the data store (e.g., Data store).

**RQ16.** The server (e.g., Registry) should perform operations to hide/unhide data when they are stored/retrieved to/from the data store (e.g., Data store).
Outline

• What are requirements?
• What are security requirements?
• Security requirements modelling?
• SREBP
• Patterns and Examples