Analyzing and Specifying Security Requirements in Early Stages of Software Development Life Cycle

Elena Simona COLES
IT&C Security Master
Department of Economic Informatics and Cybernetics
The Bucharest University of Economic Studies
ROMANIA
simonacoles@yahoo.com

Abstract: As of lately, more and more studies show that, at a global level, one of the most important causes of economic loss is unauthorized access to informatics resources. For example, a CSI/FBI Computer Crime and Security report from 2006 states that, in the top identified causes for economic loss in the USA, unauthorized access to informatics resources takes second place (Gordon et. al., 2006). Without a doubt, given the accelerated growth of software industry in the present, this need of security should be taken into account in Romania as a top priority. At the moment, in the requirements extraction phase from the software development life cycle, importance is given only to functional requirements. Thus appears the problem of identifying adequate solutions for treating non-functional requirements, where security requirements are included.

Key-Words: Security, Software, Life Cycle, Development

1. Introduction

Romanian software developers and software related service providers have already started taking measures to obtain a higher level of security for their products, but not nearly enough. These measures are included in the later stages of development, instead of being included in the requirements extraction phase. The greatest issue concerning high-risk systems appears because of the lack of detailed knowledge about software security the developers of these systems have. This isn't an issue to be underestimated, because the security of said systems isn't compromised by conventional methods like breaking mechanisms as encryption or security protocols, but exploiting weaknesses and vulnerabilities that are found after the system has been launched. 

Non-functional requirements, especially security requirements, are postponed after the analysis and design phase and are being treated as a low priority in a later phase of the project's implementation. This delay often leads to a compromise: buying a standard security solution from a specialized vendor, instead of creating a dedicated, custom one that could be better adapted to the system's context.

Recent studies lead us to believe that it is to be desired for software developers or software-related services suppliers to take into consideration security aspects from the very beginning of the software development life cycle, to be exact: starting from the analysis and design phase. Doing this, it will be easier and more effective to identify security flaws or vulnerabilities before the solution is implemented. It would also greatly reduce both time and financial costs. Furthermore, business analysts can identify security constraints and possible security non-compliances from the very beginning, given their direct and close link to their target domain's experts.

Thus, correctly documenting security requirements may bring great benefits to the final product or service. Given the aforementioned reasons, in this paper you will find techniques for documenting, analyzing and modeling security requirements, chosen so that they would be easy to apply and understand by business analysts and implementation consultants.

Given the present context of the Romanian software market, in which just meeting the
deadline for an analysis and design deliverable is a real challenge, the following considerations may prove useful, despite being more time-consuming.

2. Problem Definition
2.1 Recent Studies and Statistics about Main Software Threats

A study made in late 2014 about the existing threats in the interconnected world of Internet shows that the total number of security incidents detected by respondents climbed to 42.8 million in the referenced year, an increase of 48% over 2013 [2]. If you do the math, this is translated in 117,339 security incidents reported per day, every day. Not to mention that many organizations do not report detected incidents to keep up appearance and a lot of security breaches aren’t discovered yet.

This means that there are still a lot security incidents undiscovered. Recently, a specialized enterprise made a survey which stated that 71% of compromises go undetected [3]. This high percentage may arise because security requirements are overlooked during the analysis and design phase of software development lifecycle.

Even though hackers or organized groups are the villains that everybody love, a big threat that has to be mentioned is represented by employees. According to a survey made by PwC in 2013 [4], 36% of security breaches are caused by organizations’ employees. In this area, there is a need in finding approaches to increase information security policy compliance, as well to define a well based role based access control to the firm’s resources.

Everyone may think that large organizations are the main target of attackers, because they deal with a lot information like: personal details of citizens, large details of consumer data, trade-strategy documents, but medium and small firms are aimed as well.

Due to the sensitive data managed, larger companies started taking care of their security mechanisms and improved their defense against cyber security threats.

This is the main explanation for the 64% jump in security incidents detected by small and medium size companies. Small organizations often consider that threat actors aim only big companies, and don’t invest in information security and implementing a security procedure or protocol for exchanging secure data. This is why, customers of small and medium revenues do not emphasize the need of security in the specification document of the software they ask for.

2.2 The Role of Business Analyst in Security Requirements Engineering

In order to respond to customer specifications, most software developers didn’t focus on security aspects. The main objective in the last years has been implementing as much functionality as possible before the deadline, and preparing next releases that fix the remained unsolved bugs [5].

It is a known fact that the software engineering community is slowly beginning to realize that information security is also important for software whose primary function isn’t related to security [6].

“There is a lot of uncertainty in return on investment for security. Companies often do not know if there are doing a good job” [7]. This uncertainty reveals the need for applying mechanisms for security requirements elicitation in order to facilitate understanding both from the business representative side and from regular software developers’ side.

To achieve this goal, of getting to the same common objective of implementing organization adapted security mechanisms, the role of liaison may be fulfilled by a business analyst, or a software implementation consultant.

The main reasons why a business analyst is suited for the role, as mentioned by Khaled M. Khan, are listed below:

- Firstly, business analysts have first-hand access to the process domain. While identifying business processes, security aspects may well be captured along. Domain experts can also provide in business process description valuable input regarding security
sensitivity and importance from the users’ perspective, not to mention constraints about their activity, policies, rules and procedures. Skipping this opportunity may dramatically hinder the adequacy and completeness of security requirements at the later stages, especially in the implementation phase;

- Secondly, during design and implementation, the models get more detailed and grow in complexity. It becomes even more challenging to identify security functions accurately in these stages of project life cycle. The team loses focus on the big picture, and the chances of overlooking certain security requirements increases;

- Thirdly, tackling security at a later stage leads to a number of compromises. Often software projects are under time pressure, resource constraints and delays to the project by fixing belated security fixes will obviously cause problems on the budget. This makes it even more likely that the final software system will contain a lot of security holes [8].

3. Security Requirements Elicitation Techniques

3.1 Introduction in Security Requirements Elicitation Techniques

After a period of study and documentation about security requirements elicitation, one can discover that there are a lot of techniques that can be applied even by a business analyst with experience only in the field of gathering and documenting functional requirements. For this purpose, a technique is a sequence of steps that lead to the elicitation of security requirements [Tondel, 2008]. This section focuses on approaches in security requirements elicitation and modeling. They are proposed based on the main threats that were identified in the latest studies on cyber security and that are convenient enough for average software developers to implement. The current literature about security requirements is filled with techniques and proposed approaches, but few offer concrete advice for immediate implementation. Even if implemented, a clear fact is that there cannot be a technique that helps a business analyst assess all security requirements. Furthermore, in the development stage, developers may add new features to improve security level, based on their experience.

3.2 Identify the Customer’s Need for Security

The customer’s need for security can be determined by analyzing the specification documentation and the organizational context for the software to be. The proposed technique for identifying the need for security is by filling a checklist with the following structure (the checklist is presented in the Appendix):

- General application data
  - Type of application – if there is a business based application or is an internal IT application, not used in business lines;
  - Type of users - what type of users will interact with the application;
  - Developer of application – if there is an internal team of developers / external team developers / internal and external team of developers;
  - Sensitivity of application – sensitivity of application from business point of view. Example: How much time the application can be offline (< 6H; <8h; <24H);
  - IT Type of Application client - type of client application: web based, client based, no user interface;
  - Business lines affected (impact) - level of impact on others business processes;
  - Need to modify existing policies, procedures? – if the new software modifies the existing policies or procedures of the customer organization;
  - Need to add a new policy or/and procedure? - if the new software
adds new policies or procedures for the customer organization;

- **Data/ Information**
  - *Data sensitivity used in application* – level of data sensitivity processed in application (confidential data, personal data);
  - *Import data from production environment* – if the new software will import data from existing environment;
  - *New data to be input/ transferred by the new application/ development* – if the new software will be used to apply existing business processes and data flow;
  - *Data stored must be protected?* – if the archiving procedures state data protection.

- **IT Infrastructure**
  - *Level of IT impact on current Infrastructure*;
  - *Need a new testing environment*;
  - *Replication/synchronization with other applications*;
  - *Interface with other applications*;
  - *Does the infrastructure need Radio communication*;
  - *External Party/ Vendor/ Outsourcing*;
  - *Are data transmitted to vendor*;
  - *Are data transmitted to other parties*;
  - *Vendor access to Organization's IT resources*;
  - *Vendor application services*;
  - *Auditing*.

It’s not recommended to use the same checklist for every type of implementation. In order to fulfill this recommendation, in the current context of software development where there are requests for new software, or for change implementation in an existing software product. Taking this reason into account, new software implementation and change requests have to be treated separately in identifying the need for security.

### 3.3 Identify Policies, Standards, Legislation and Techniques That May Apply

After analyzing the organizational context, if security engineering is proved to be a real need, the next step is to gather as many security requirements from the organization’s artifacts as possible. The business analyst has to look for:

- Policies;
- Legislation;
- Standards applied by the domain experts;
- Identify the most important assets that need protection – it is recommended to analyze the assets determined from multiple points of view: customer, user, owner, even attacker, because every stakeholder has a subjective opinion on security related assets. An asset that may seem sensitive to an interviewed stakeholder, may not seem so important to others. In order to protect the most sensitive assets, it is advised to prioritize them after the brainstorming or interview sessions [9].

![Figure 32. Modeling framework for secure business process elicitation](image)

Security Policy refers to rules that the organization asks for in the development stage of a secure system. These rules and procedures define the business transactions that carry security-sensitive actions, like any interaction with sensitive records (read, delete, modify).
A real problem about security requirements is that in the specification document or other artifacts, they are often formulated in a way that does not facilitate acceptance testing after the implementation.

Example of abstract formulations: "The system should be secure".

Donald G. Firesmith specifies a method to formulate a reusable template for security requirements [10]:

"The [application/component/data center/business unit] shall protect the data it transmits from corruption (e.g. unauthorized addition, modification, deletion or replay) due to [unsophisticated / somewhat sophisticated/ sophisticated] attack during execution of [a set of interactions/ use cases] as indicated in the [specified table]”, where the ‘specified table’ refers to a table of user stories or use cases with their minimum acceptable level of security.

Taking into account recommended templates to formulate a security requirement or function, once they are identified, they are linked with their corresponding business process (functional requirement) and both executed.

3.4 Incorporating Security Requirements in Functional Requirements Documentation

After having all security requirements in a traceable form during the software development lifecycle, the next step is to involve security aspects in early stages of information system development. In this way, analysts become more aware of possible threats and violations by involving security aspects in business process [11]. The viewpoint considering this approach is based on a socio-organizational perspective, as researchers move towards the technical security details [12].

As User stories and Agile methodology are still in an incipient stage in Romanian software development companies, the proposed technique is the extended UML-Use Case, to involve security in the analysis and design phase of software development lifecycle.

Using Enriched use case and activity diagrams to emphasize security requirements

1. Enriched Use case

The main focus on use cases refers to what the system does, and not how it does the specific activity. Enriched use case incorporates security constraints, security subjects and security actors into the design of information systems [13].

| Table 1. Enriched Use-case proposed [Siponen, 2006] example with bolded security details |

<table>
<thead>
<tr>
<th>Use case: Use case name (codification)</th>
<th>Version: version number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Summary/Purpose: What is the purpose of this use case?</td>
<td></td>
</tr>
<tr>
<td>Priority: High / Medium/ Low</td>
<td></td>
</tr>
<tr>
<td>Actor/security subject: User and user role in the system (=security subject)</td>
<td></td>
</tr>
<tr>
<td>Security classification of the subject: Top secret / Secret / Confidential / Public</td>
<td></td>
</tr>
<tr>
<td>Preconditions: What are the preconditions for this use case? What must be fulfilled in order to run this use case?</td>
<td></td>
</tr>
<tr>
<td>Result: What is the result of this use case?</td>
<td></td>
</tr>
<tr>
<td>Definition: The main success scenario of this use case</td>
<td></td>
</tr>
<tr>
<td>Processed information: The information entity processed in this use case (=security object)</td>
<td></td>
</tr>
<tr>
<td>Classification of the object: Top secret / Secret / Confidential / Public</td>
<td></td>
</tr>
<tr>
<td>Exceptions: Define alternative exceptions and scenarios</td>
<td></td>
</tr>
<tr>
<td>Security policy/Specific security restrictions: The security policy regarding this use case</td>
<td></td>
</tr>
<tr>
<td>Requirements: Links to specified requirements of this use case</td>
<td></td>
</tr>
<tr>
<td>Test cases: Links to specified test cases of this use case</td>
<td></td>
</tr>
</tbody>
</table>

2. Activity Diagram

Activity diagrams are UML’s most used modeling method in business analysis field. They emphasize business processes and data flow in the organization showing:

- Actors involved in the business process;
- Sequential activities realized by the actors involved;
- Flow of documents and data in the studied activity.

Table 2. Comparison of best known security requirements modeling techniques

<table>
<thead>
<tr>
<th>Security modeling approaches</th>
<th>Attack Security mechanism</th>
<th>Vulnerability</th>
<th>Assets</th>
<th>Functional requirements</th>
<th>Security requirements</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack Tree [14]</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modeling and understanding attacks, identifying vulnerable points</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fault Tree [15]</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Shows likelihood of the system failures. Unlike the attack tree, this approach is more difficult to adapt with.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attack graph [16]</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Attack models and analysis of computer networks to identify vulnerabilities. Attack graphs cannot express the impact of attacks on system functionality. They define the steps, preconditions and post-conditions of an attack.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abuse case [17]</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This method is recommended when there is a need for communicating the possible flaws associated with use cases to customers or end users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Misuse case [18]</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>With misuse case, a business analyst can represent the behavior that the system does not want to occur. Useful to model attacks in conjunction with functional requirements for security requirements elicitation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended Misuse case [19]</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Are used to model vulnerabilities to identify all possible attacks and threats. Uses inverted actors and inverted use cases. It also includes insider vulnerabilities.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CORAS Risk Modeling Notation [20]</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Represents model based risk analysis. Recommended because it is applied on all concepts.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secure Tropos [21]</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Goal based technique for security requirements engineering.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMLsec [22]</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Used to extend UML diagrams by applying security aspects. Uses stereotypes and tags.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secure UML [23]</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model driven development of secure systems based on UML. Are used to generate access control concepts.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note! N indicates that the approach does not consider the concept in the corresponding column; Y indicates that the approach considers the concept in the corresponding column*
4. Research Details and Future Plans

This paper proposes a checklist in order to identify in a few steps the need for security of a new system, or a change request for an existent system. This checklist specified above, among other techniques identified that may easy apply in nowadays development of information systems. As well, a checklist is proposed in the appendix for exemplification.

How was the checklist created?
1. Firstly, I researched the main threats that endanger information system lately;
2. Then, based on the main threats identified at step 1, I asked around some business analysts and business analyst team leaders from my company about criteria that define the need for security;
3. I gathered all the information and criteria and filtered them afterwards;
4. Every criteria was transformed into a single choice question with 2 or many possible answers;
5. With all the info gathered at step 4, I ordered the questions first on 4 categories that will contain the questions (General application data / Data and information / IT Infrastructure/ External Party/ Vendor / Outsourcing), and then on security level importance (Low / Medium / High/ Very High);
6. For each question, considering the security importance identified at step 4, I accorded some points ordered by security importance;
7. Then I identified the criteria that will determine the need for security considering a level of points that can be gathered base on step 6 and based on the answers;
8. If there a need for security is identified at step 6, taking into account the number of points based on the answers, the level of security recommended is displayed as: Very High, High, Medium or Low.

Future plans for this research are to extend the checklist in order to display a requirements elicitation technique based on the organizational context and project scope.

5. Conclusion

Incorporation of security requirements in business process modeling essentially improves secure software engineering. In Nagaratnam et al. (2005), the authors argue that business process modeling is an ideal time in the life cycle of software system development to begin capturing the business security requirements that address security concerns related to the business. This argument of incorporating security early into the business specifications has been consistently raised, supported, and advocated by security researchers and business analysts.

The techniques proposed in this paper do not focus on the formally pleasing or academically correctness. Taking into account current methods for functional requirements used by business analysts, these lightweight approaches are proposed by looking at what can be put in practice in security requirements engineering. Teaching old dogs new tricks is somewhat difficult, and if the techniques aren’t easy to understand and use, they will remain in books and never be applied.

Acknowledgement

Parts of this paper were presented at The 8th International Conference on Security for Information Technology and Communications (SECITC 2015), Bucharest, Romania, 11-12 June 2015.

References