@incollection {springerlink:10.1007/978-3-642-29231-6_6,
author = {Herrmann, Andrea and Morali, Ayse and Etalle, Sandro and Wieringa, Roel},
title = {Risk and Business Goal Based Security Requirement and Countermeasure Prioritization},
booktitle = {Workshops on Business Informatics Research},
series = {Lecture Notes in Business Information Processing},
editor = {Niedrite, Laila and Strazdina, Renate and Wangler, Benkt and Aalst, Wil and Mylopoulos, John and Rosemann, Michael and Shaw, Michael J. and Szyperski, Clemens},
publisher = {Springer Berlin Heidelberg}, pages = {64-76}, volume = {106}, url = {http://dx.doi.org/10.1007/978-3-642-29231-6_6}, year = {2012}}

Risk and Business Goal Based Security Requirement and Countermeasure Prioritization

Andrea Herrmann[1], Ayse Morali[2], Sandro Etalle[3] and Roel Wieringa[4]

[1] Independent researcher, AndreaHerrmann3@gmx.de
[2] Ascure N.V., St. Denijs-Westrem, Belgium, Ayse.Morali@ascure.com
[3] Eindhoven Technical University, Eindhoven, The Netherlands, s.etalle@tue.nl
[4] University of Twente, Enschede, The Netherlands, roel.wieringa@utwente.nl

Abstract. Companies are under pressure to be in control of their assets but at the same time they must operate as efficiently as possible. This means that they aim to implement "good-enough security" but need to be able to justify their security investment plans. Currently companies achieve this by means of checklist-based security assessments, but these methods are a way to achieve consensus without being able to provide justifications of countermeasures in terms of business goals. But such justifications are needed to operate securely and effectively in networked businesses. In this paper, we first compare a Risk-Based Requirements Prioritization method (RiskREP) with some requirements engineering and risk assessment methods based on their requirements elicitation and prioritization properties. RiskREP extends misuse case-based requirements engineering methods with IT architecture-based risk assessment and countermeasure definition and prioritization. Then, we present how RiskREP prioritizes countermeasures by linking business goals to countermeasure specification. Prioritizing countermeasures based on business goals is especially important to provide the stakeholders with structured arguments for choosing a set of countermeasures to implement. We illustrate RiskREP and how it prioritizes the countermeasures it elicits by an application to an action case.

Keywords: Non-functional requirements; Risk assessment; Misuse Cases; IT architecture; Security; Prioritization.

1 Introduction

Today, organizations are under high pressure to prove that they are in control of their assets, which means among other things that they must prove that they sufficiently secured their IT assets. At the same time, they are increasingly cost-sensitive and hence they aim at reducing security risks in a cost-effective way. The common solution is to use checklists to identify the largest risks and mitigate them. However, checklists are based on past experience and are useful for achieving consensus among experts, but do not necessarily provide justifications that are based on business goals or technical characteristics of the system. Such ad hoc analyses are risky in the face of current fast-changing information technology (IT) [14, 20]. Furthermore, such justifications provide a proof of common maturity level which is necessary for

networks of businesses to operate securely and effectively. In a previous work we presented RiskREP [8]. RiskREP allows the justification of security investments in terms of the vulnerabilities of the business processes and the IT architecture in relation to the business goals to be achieved.

We build on current proposals for extending requirements engineering (RE) methods with security risk assessment (RA) [4, 6, 7, 13, 15, 18, 19]. In Section 2, we compare some RE methods and RA methods to necessary requirements elicitation and prioritization features. We present the meta model of RiskREP in Section 3, present how RiskREP elicits and prioritizes countermeasures by linking business goals to countermeasure specifications in Section 4, and discuss lessons learned from an action case study in Section 5.

2 Related Work

In this section, we compare some well-known RE and RA methods based on their requirements elicitation and prioritization properties. Tables 1 and 2 present an overview of this comparison. Please note that this list cannot be complete, considering the vast amount of existing methods. Here, we present only those methods that satisfy most of the properties which we considered as success criteria when developing RiskREP. Tables 1 and 2 use these properties as criteria for comparing the methods.

We advocate that the elicitation of security requirements must follow a systematic process, because this supports the traceable justification each requirement. In order to be complete, we want to differentiate between business and quality goals, to consider both permissible use and misuse, and to explicitly include different stakeholder views in order to arrive at security requirements which reflect the multi-perspective nature of security.

When prioritizing requirements and identifying the optimal set of security requirements to implement, one needs to know the risks against which the requirement will counteract. Risk is described by impact and incident likelihood. Security requirements are compared to each other both based on their monetary costs and effectiveness against the risk, i.e. risk reduction achieved. Additionally, combined effects of requirements play a role, like the potential of security measures to replace each other or to complement each other.

To systematically elicit security requirements, Elahi and Yu [5], Stamatis [18] and Mayer et al. [12] propose to derive requirements from high level goals. . We believe that a security requirements elicitation method should also differentiate between business goals (i.e. desired properties of the business) and quality goals (i.e. desired properties of the software) – where quality goals include security goals). Despite the fact that most of the approaches that we compare, e.g. [6, 11, 15, 19] differentiate between functional and non-functional goals of software systems, none of them differentiate between business and quality goals.

To address the security concerns of system owners, recently developed RE methods, e.g. [5, 9, 17, 19], model not only permissible uses but also misuses of system components.

Eliciting information on permissible uses and misuses, on business goals as well as quality goals, requires expertise of stakeholders with different backgrounds. Only a few of the approaches that we consider in this comparison ([2, 5, 9, 11, 17]) express how different stakeholder views can be considered when eliciting information. GSRM [9], for instance, differentiates the perspectives of user, business analyst, requirements engineer, and risk manager.

Once the security requirements are identified, one has to check whether they are implementable within the available budget. Usually, this is not the case, and one has to decide which set of requirements should be implemented and which requirements can be disregarded. Making such a decision requires the estimation of the security risks the system is exposed to, considering the trade-off among the different requirements, as well as their costs and effectiveness. However, only some methods (such as FMEA [18], Tropos based approaches [1, 5], GSRM [9], Attack Graphs [16], extended KAOS [19], and the approach proposed by Mayer et al. [12]) take into consideration the risk the system is exposed to.

Table 1: Comparison of some RE methods with respect to requirements elicitation and prioritization features.

	Elahi and Yu [5]	Misuse Cases [17]	extended KAOS [19]	ATAM [11]	NFR frame-work [15]
Requirements elicitation					
Systematic process	derives soft-goals from goals	yes	no	no	derives soft-goals from goals
Differentiation between business and quality goals	goals and soft-goals	no	functional and non-functional goals	yes	technical and business objective
Considering both permissible use and misuse	use and misuse	use cases & misuse cases	goal and anti-goal	no	no
Considering different stakeholder views	yes	yes	no	yes	no
Requirements prioritization					
Estimation of impact	no	no	no	no	no
Estimation of incident likelihood	level of evidence	no	for deter-mining the granularity	no	no
Prioritization	yes	real cost	no	volume of	no

based on monetary costs of requirements				change	
Considering effectiveness of requirements	3 levels	no	no	no	no
Considering combined effects of requirements	between soft-goals	no	no	trade-off points	between soft-goals

Table 2: Comparison of widely known RA methods with respect to requirements elicitation and prioritization features.

	FMEA [18]	Attack Graphs [16]	CORAS [2]	Secure Tropos [1]	GSRM [9]
Requirements elicitation					
Systematic process	yes	no	no	yes	yes
Differentiation between business and quality goals	no	no	no	3 layers: asset, event, treatment	project goals and sub goals
Considering both permissible use and misuse	no	no	no	yes: tasks and risks	risk events and tasks
Considering different stakeholder views	no	no	yes	no	yes
Requirements prioritization					
Estimation of impact	failure effect	no	depends on selected model	severity of impact	risk impact
Estimation of incident likelihood	occurrence of failure	probability, average time or cost/effort	depends on the model	event likelihood	risk likelihood
Prioritization based on monetary costs of requirements	no	financial loss or loss of system	no	yes	no
Considering effectiveness of requirements	detection rate	no	no	qualitatively	effectiveness
Considering combined effects of requirements	no	no	no	qualitatively	no

The methods that take into consideration effectiveness levels of requirements refer to different attributes of the IT system that is analyzed. Elahi et al. [6] differentiate among three levels according to whether the countermeasure alleviates the effects of vulnerabilities, patches them or prevents malicious tasks. Secure Tropos [1] differentiates between four categories of countermeasures (removal/avoidance,

prevention, attenuation, and retention) depending on how they mitigate the risk in the event layer. Finally, FMEA [18] differentiates according to incident detection rate.

When taken together, requirements may contradict with each other or support each other. Elahi and Yu [5], NFR framework [15], Mayer et al. [12], and Secure Tropos [1] consider these combined effects and prioritize the requirements accordingly. ATAM [11] also considers how requirements affect each other "trade-off points".

Tables 1 and 2 show that the RE and RA methods we have compared do not have all the requirements elicitation and prioritization features that we think are important. Therefore, we developed RiskREP [8]. RiskREP is built on the CRAC++ [14] and MOQARE [7] methods. The following Section 3 presents RiskREP´s meta model which explicitly considers different perspectives on security, and Section 4 illustrates RiskREP´s systematic process with extracts from an action case.

3 Meta model of RiskREP

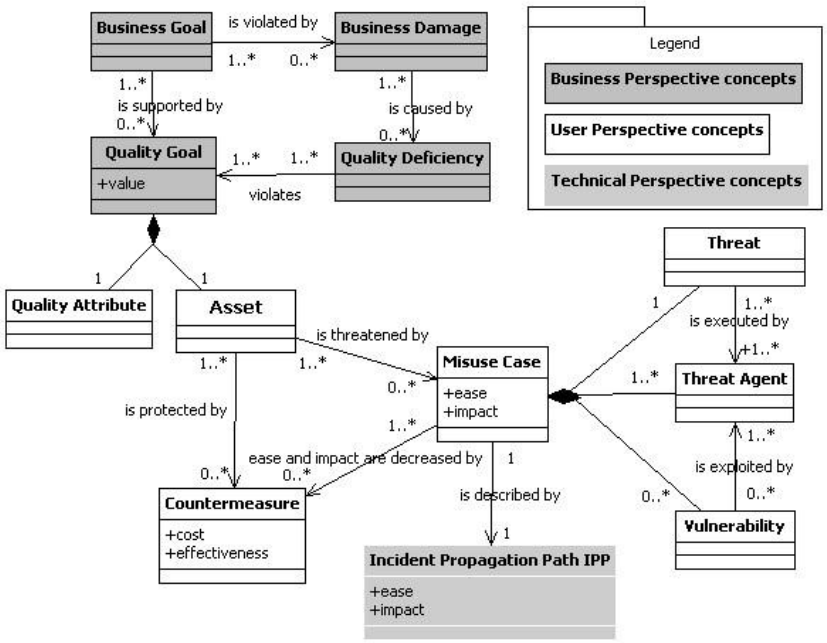

Fig. 1: Meta-model showing the concepts and their interrelations.

The meta model (Figure 1) contains concepts from three perspectives, i.e. the *business perspective*, the *user perspective* and the *technical perspective*. Before RiskREP is applied, a model of the system´s architecture and specifications of the system´s functionality from user perspective (e.g. modeled as use cases) must exist. To these system models, RiskREP adds the security aspect.

Business perspective: *Business goals* are desired properties of the business. Business goals justify system requirements. An example of business goal is "efficient business processes". A *business damage* is a state or activity of the business that violates a business goal. The business damage completes the business view by asking what should not happen. An example of business damage is "users don't use the system to be". A *quality goals* are desired qualities of the IT system, i.e. a desired state of the system. These goals are expressed as high-level quality requirements that consist of a quality attribute and an asset, like "confidentiality of password". A *quality deficiency* is a lack of quality attribute for an asset that violates quality goals and might causes business damage.

User perspective: Quality attributes are attributes of the system to be protected. They describe aspects or characteristics of quality, e.g. confidentiality. We use the quality attributes of the ISO 9126 [3] and assume that these completely categorize all relevant aspects of an IT systems quality. Assets are parts of the system that are valuable for the organization, e.g. information, software, or hardware. They need to be protected from malicious activities in order to achieve business goals. Value quantifies the criticality of each quality goal with respect to the business. The value is used to prioritize the quality goals against each other. It is determined by the impact that the compromise of an asset would cause to the business.

Misuse Cases [17] describe scenarios in which a threat agent can cause a quality deficiency. The misuse case takes the perspective of the user and describes what happens at the interface between user and system. They are identified by analyzing the business process and the Use Cases of the system. The misuse cases are prioritized based on their execution ease and the impact, which they cause to the asset(s). *Threats* are actions, which cause a quality deficiency that causes the violation of a quality goal, e.g. data theft violates the confidentiality of data. *Vulnerabilities* are a property of the assets or the IT system or its environment that can be exploited by threat agents. This exploitation could violate a quality goal. Vulnerabilities can be unwanted properties like "lack of technical change management" or also wanted properties of the system such as "Single-Sign On". A *threat agent* is a person, i.e. an insider or an outsourcer or an outsider that intentionally or unintentionally executes a threat. A threat agent can be characterized in terms of his motivation, goal and attributes, e.g. disgruntled employee.

Countermeasures are mitigation, detection or prevention mechanisms. They partly or completely counteract a threat-vulnerability pair or the threat agent, and reduce the estimated impact at threat/vulnerability and/or the ease of threat execution. Countermeasures are expressed as (security) requirements on the IT system. Cost is an attribute of a countermeasure. It consists of implementation cost and the cost of ownership. Depending on the depth of the assessment we either use partially ordered scale or the real costs. In case the real costs are used then the risk expert may calculate the implementation cost based on required hours and salary per hour. The *expected effectiveness* of a countermeasure is given by the expected risk reduction it achieves. Most countermeasures either influence the impact or the execution ease of an Incident Propagation Path.

Technical perspective: *Incident Propagation Paths* are descriptions of misuse case from the technical perspective. In some cases, an Incident Propagation Path consists of several interconnected steps. That is a threat agent causing a quality deficiency on an asset by executing one or more threats, which exploit vulnerabilities of several assets. Such Incident Propagation Path scenarios are important for humans to imagine the flow of events including the causes and consequences of incidents. Like the misuse cases, the Incident Propagation Paths are prioritized based on their execution ease and the impact they have. There may be several Incident Propagation Paths realizing the same misuse case. The *execution ease* of a misuse case is an estimation of the effort required to carry out a misuse case. This effort is determined by the most resistant vulnerability that needs to be exploited to carry out the misuse case. In our approach, the execution ease is considered to be in correlation with the likelihood that a threat is actually executed by the "strongest" threat agent. *Impact* is the damage caused to the assets by the execution of a misuse case.

4 Steps of the RiskREP method

The four steps of the method are:

1. *Quality goal analysis:* identify business goals, business damages, quality deficiencies and quality goals;
2. *Risk analysis:* identify misuse case (threats, threat agents, vulnerabilities) and estimate their impact on assets, and their ease of execution by means of incident propagation paths;
3. *Countermeasure definition:* specify countermeasures and estimate their cost; and
4. *Countermeasure prioritization:* assess effectiveness of countermeasures in reducing misuse case risk, their cost and dependencies.

At each of these steps, it is possible to either analyze the complete system, all business goals, and all misuse cases, respectively or to focus on the most important aspects. RiskREP is currently supported by spreadsheet tables.

The information that the RiskREP method uses is elicited from three stakeholder categories: business owner, IT manager and security officer who represent the business, IT and user perspective, respectively. The method is executed by an RE expert and a risk expert, who elicit the necessary information by semi-structured interviews with the other stakeholders. We applied the method in the TUgether project of the University Braunschweig (TU), in which a portal is developed to provide all on-line services of the TU, such as email, library access, registration for exams etc. available to students and employees. The portal must allow students to sign-on via one individually configurable interface. One major objective is that all students should eventually use the portal.

In the first phase of the project the portal framework product was selected which satisfied requirements best. Eighty functional and non-functional requirements were specified and about 70 products were considered. Our case study is restricted to the eleven security requirements of the 80 requirements.

The TUgether project was at an early development stage at the time we started applying RiskREP to it. We received from the project team the complete requirements specification. After analyzing it, we had several meetings with the project team to elicit the information RiskREP uses, such as the IT architecture of the TUgether portal. We concluded the action case by presenting the output of the method to the business owner, IT manager and security officer in a meeting and asked their opinion about the information RiskREP delivered. We now run through the steps of the method.

Step 1: Quality goal analysis

We could infer the security-related business perspective concepts from a project report which had been written before the case study. Figure 2 shows an extract of this analysis. Business goal "gaining user acceptance" (BG5) is threatened by one business damage, "Portal will not be used" (BD6). Three quality deficiencies may cause this, viz. User unfriendliness (QD7), lack of trust (QD8), and lack of added value (QD9). Because of the scope of our case study, we analyzed only quality goal "lack of trust" (QD8) further. QD8 can be avoided by three high level quality goals, i.e. Confidentiality of assets (QG5), Integrity of assets (QG6), and Availability of assets (QG7). Step 1 ensures that all software quality goals are justified by to business goals – including security.

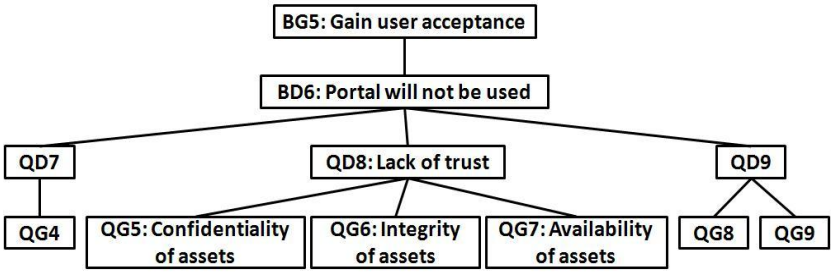

Fig. 2: Business concepts elicited with RiskREP

Step 2: Risk analysis

The risk expert first identifies possible misuse cases that may threaten a quality goal and estimate their impact on assets and ease of execution. In addition, the security expert draws Incident Propagation Paths through the architecture that connects entry points of the system to the misuse case. This allows us the estimation of the ease of execution of the misuse case. Modeling the execution ease is also the main difference between Incident Propagation Paths and Misuse Case Maps [10].

The risk expert also assesses the value of each quality goal, for example by using value models for availability [20] or confidentiality [14] and then estimates the impact or damage caused by the misuse case to these quality goals. This way we maintain the link between business goals and impact of a misuse case.

For example, in the case study, misuse case "Manipulation of account data" (MC5) threatens quality goal "Integrity of assets" (QG6). There are five threat agents, viz. user, hacker, portal admin, portal developer and service developer. In the portal architecture (Figure 3), the critical IT assets related to misuse case "Manipulation of data" (MC5) are: TUgether portal server, LDAP server and Development server. We used a scale from 1 (low) to 3 (high) to indicate execution ease and impact. The execution ease of misuse case "Manipulation of data" (MC5) was estimated 1.5 and its impact was estimated 1. Incident Propagation Paths are described by the misuse case good enough here and therefore we did not draw them. In total, related to quality goal "Integrity of assets" (QG6), we identified ten misuse cases, one of which we show in Table 3. As this table illustrates, the risk of a misuse case is represented by a pair (ease of execution, impact on assets) where each of the two components of risk has a totally ordered scale. This defines a partial ordering of misuse cases according to their risk. Unlike in other risk assessment methods, we do not multiply ease with impact, but instead form categories of misuse cases, based on the priorities of the stakeholders. For instance, an misuse case with ease and impact equal to 3 can be called a "catastrophe", and the misuse case category "frequent, but harmless" describes misuse case where ease is high, but impact is low.

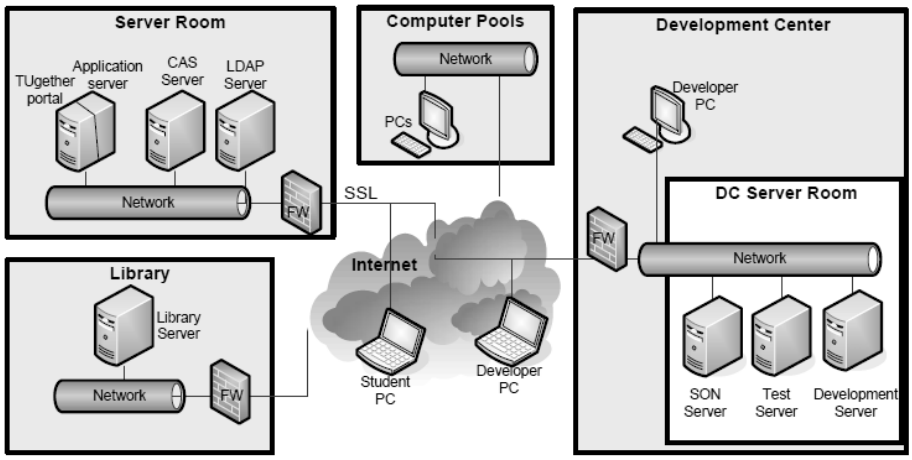

Fig. 3: TUgether portals IT architecture. (FW: Firewall, DC: Data Center, CAS: Central Authentication Service, SON: Personal Development Server.)

Step 3: Countermeasure definition.

The security officer and RE expert compose a set of countermeasures by taking them from existing checklists. These checklists are part of RiskREP and contain general

countermeasures for 167 threat vulnerability pairs. In this step of RiskREP, one brings these general measures to a concrete, realizable level by specifying which component each of them applies to and how. Table 2 shows the results of this step on our case. Cost estimations are indicated by a 0 (no cost), 1 (changing the settings of applications), 2 (installing and maintaining freely available countermeasures) and 3 (purchasing, installing and maintaining countermeasures).

Table 3: Some misuse cases (MC) and their attributes.

MC ID	**risk (ease, impact)**	**Threat agent**	**Threat**	**Vulnerability**
MC5: manipulation of account data	(1.5,1)	Hacker	data get lost or are manipulated during transfer	Portal does not manage data and therefore data synchronization between portal and services is necessary
MC9: no logout in computer pool	(1,3)	User	does not log out after having used the portal on a computer in the public computer pool	no access control to computer pools

Step 4: Countermeasure prioritization

By applying countermeasures to misuse cases, one reduces risk. However, applying countermeasures usually means increased spending. Therefore, RiskREP aims at finding the ideal set of countermeasures to be applied. The best set of countermeasures is that with minimum total cost and maximum risk reduction. To find an optimum set, we must compare several sets of countermeasures. In practice, the security budget of the system is often the main delimiter for the ideal set of countermeasures. To prioritize countermeasures, their effectiveness in reducing the risk of misuse case must be quantified. We measure the effectiveness of a countermeasure with respect to a risk by the effect on decreasing both the ease of an attacker executing an attack and the impact of that attack. Ease as well as impact can be increased (+1), decreased (-1) or unaffected (0 points) by the application of a countermeasure. In this way it is easy to estimate and is less prone to mistakes. If necessary, RiskREP allows using more sophisticated scales.

Countermeasures interact with each other. For instance, some may be overlapping, or diminish each other's effectiveness. We documented the combined effect of pairs of countermeasures for TUgether in a two dimensional matrix containing 10 interactions, and discussed this with the security officer. The matrix is sparse and not symmetric; because it is possible that countermeasure c_1 influences c_2, but not vice versa. In the case study, it contains 10 interactions, whereas among the 10 countermeasures 90 different interactions would be theoretically possible.

We then prioritized countermeasures according to their cost and effectiveness. Just as for risk, no multiplications or additions can be done because the scales we use are ordinal. The security objectives of companies and their security strategies differ from each other. Therefore, RiskREP defines company-specific heuristic for the countermeasure prioritization. We classified countermeasures according to their cost and effectiveness in the following categories:

- *no effect:* both execution ease and impact of a misuse case are not modified by the countermeasure
- *contra-effective:* both execution ease and impact of a misuse case are increased, or one is increased and the other one is not modified, by the countermeasure;
- *counter-effective:* The countermeasure increases execution ease and reduces impact of the misuse case, or vice versa;
- *low hanging fruit:* cost is 0, either only execution ease or only impact of a misuse case is reduced by the countermeasure; or both execution ease and impact of a misuse case are reduced by the countermeasure;
- *cost-efficient:* cost is 1 and either only execution ease or only impact of a misuse case is reduces by the countermeasure; or both execution ease and impact of a misuse case are reduced by the countermeasure;
- *cost-effective:* cost is 2 and both execution ease and impact of a misuse case are reduced by the countermeasure;
- *expensive:* cost is 2 or above and either only execution ease of a misuse case is reduced by the countermeasure or only impact of a misuse case is reduced by the countermeasure;
- *expensive effectiveness:* cost is 3 and both execution ease and impact of a misuse case are reduced by the countermeasure.

To choose the optimal set of countermeasures, we did not use a formula which optimizes the systems added value automatically, but rather decided for a countermeasure selection strategy together with the stakeholders. In this case, the strategy is on countermeasure effectiveness and cost. Accordingly we suggested the stakeholder to implementing all "low hanging fruit" countermeasures. Furthermore, since defining the categories also influences the strategy, we asked for stakeholders' approval after defining them. This way of choosing the countermeasures to be implemented is a heuristical one which allows making decisions transparently and based on objective criteria, but still is simple and easy to execute.

5 Analysis and discussion

RiskREP is designed to elicit security requirements following a systematic process, and considering several perspectives of security: the business perspective, user perspective and technical perspective, and both permissible use and misuse. For prioritizing countermeasures, RiskREP considers misuse cases´ impacts and incident

likelihoods, countermeasures´ monetary costs and effectiveness against the risk, and combined effects of countermeasures. We have applied RiskREP to an action case in order to verify whether RiskREP supports security requirements elicitation and prioritization in a way that one can control whether the result is complete or lightweight.

Our action case study showed that RiskREP can be used and leads to a list of misuse case partially ordered by risk, and motivated in terms of system architecture as well as business goals. It also leads to a prioritized list of countermeasures agreed on by stakeholders. It took us about four hours to apply RiskREP to one quality goal. This is comparable to the time currently spent on security RE. So, we conclude that RiskREP can be used within the available budget for security RE.

But is it better than the method currently in use? Did it lead to a better understanding of security risk and/or to a better set of countermeasures, in terms of estimated cost and estimated effectiveness? Before we applied RiskREP, the university was using a collection of requirements grouped according to each attribute of the system. These requirements were elicited from different stakeholders, and eleven high-level requirements were about security. They were of different granularity levels, and it was neither possible to compare their risk level, nor to validate their completeness. By contrast, RiskREP systematically analyzes the risks both from user perspective and technical perspective under consideration of all use cases and data flows. We argue that this an improvement w.r.t. the previous way of working. While RiskREP potentially could elicit all countermeasures completely, at each step it is possible to focus on the most relevant aspects, e.g. most important quality goals, most important misuse case etc. and to document this decision. So, RiskREP supports also a light-weight analysis that is focused on the most important elements.

Comparing RiskREP to other security RE methods we note that we do not use our ordered scales of misuse cases (based on ease of execution and impact on assets), cost and effectiveness in inadmissible ways, such as by multiplying impact and ease of executing an Incident Propagation Path. This makes the results of using our method more meaningful than the results of other methods. Assuming that in this particular case study, RiskREP could be used and is an improvement, could it be used in other cases, too? Would other people be able to use it with the same effectiveness in other cases? RiskREP assumes that the information listed in the meta model can be elicited and that stakeholders are able to reach agreement about a countermeasure prioritization in terms of their cost and effectiveness. However, for it to be used by other requirements engineers than us, we need to supply RiskREP with tool support and supporting manuals. We are planning to develop this in the near future.

References

1. Y. Asnar, P. Giorgini and J. Mylopoulos, Goal-driven risk assessment in requirements engineering. Requirement Engineering Journal, 1-16 2010.

2. F. Braber, I. Hogganvik, M. Lund, K. Stølen, and F. Vraalsen. Model-based security analysis in seven steps — a guided tour to the CORAS method. BT Technology Journal, 25(1):101–117, 2007.
3. I.S.O. I.E. Commission. ISO/IEC 9126, Information technology - Software product evaluation - Quality characteristics and guidelines for their use., 1991. http://www.iso.org.
4. E. Dubois, P. Heymans, N. Mayer, and R. Matulevicius. A systematic approach to define the domain of information system security risk management. In S. N. et al., editor, Intentional Perspectives on Information Systems Engineering, p. 289-306. Springer, 2010.
5. G. Elahi and E. Yu. Modeling and analysis of security trade-offs - A goal oriented approach. Data Knowledge Engineering, 68:579–598, 2009.
6. G. Elahi, E. Yu, and N. Zannone. A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. Requir. Eng., 15(1):41–62, 2010.
7. A. Herrmann and B. Paech. MOQARE: misuse-oriented quality requirements engineering. Requir. Eng., 13(1):73–86, 2008.
8. A. Herrmann, A. Morali, S. Etalle and R. Wieringa. RiskREP: Risk-Based Security Requirements Elicitation and Prioritization. In: Perspectives in Business Informatics Research, 1st International Workshop on Alignment of Business Process and Security Modelling, 155-162, 2011.
9. S. Islam and S. Houmb. Integrating risk management activities into requirements engineering. In Proc. of the 4th Int. Conf. on Research Challenges in Information Science. IEEE Computer Society, 2010.
10. P. Karpati, G. Sindre, and A. Opdahl. Visualizing cyber attacks with misuse case maps. In Requirements Engineering: Foundation for Software Quality, pages 262–275, 2010.
11. R. Kazman, M. Klein, P. Clements, and N. Compton. Atam: Method for architecture evaluation. Technical Report CMU/SEI-2000-TR-004, CMU, 2000.
12. N. Mayer, E. Dubois, and A. Rifaut. Requirements engineering for improving business/it alignment in security risk management methods. In Proc. of the 3rd Int. Conf. Interoperability for Enterprise Software and Applications, page 12. I-ESA, 2007.
13. A. P. Moore, R. J. Ellison, and R. C. Linger. Attack modeling for information security and survivability. Technical Report CMU/SEI-2001-TN-001, CMU, 2001.
14. A. Morali. IT Architecture-Based Confidentiality Risk Assessment in Networks of Organizations. PhD thesis, University of Twente, Enschede, The Netherlands, 2011.
15. J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E. Yu. Exploring alternatives during requirements analysis. IEEE Software, 18:92–96, 2001.
16. C. Phillips and L. Swiler. A graph-based system for network-vulnerability analysis. In Proc. of the 1998 workshop on New security paradigms, pages 71–79. ACM, 1998.
17. G. Sindre and A. Opdahl. Eliciting security requirements with misuse cases. Requir. Eng., 10(1):34–44, 2005.
18. D. Stamatis. Failure mode and effect analysis FMEA from theory to execution. American Society for Quality Press, 2003.
19. A. van Lamsweerde, S. Brohez, R. D. Landtsheer, and D. Janssens. From system goals to intruder anti-goals: Attack generation and resolution for security requirements engineering. In Proc. of RHAS Workshop, pages 49–56. Essener Informatik Beitraege, Bd.6, 2003.
20. E. Zambon. Towards Optimal IT Availability Planning: Methods and Tools. PhD thesis, University of Twente, Enschede, The Netherlands, 2011.