RiskREP: Risk-Based Security Requirements Elicitation and 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. In this paper, we present a Risk-Based Requirements Prioritization method (RiskREP) that extends misuse case-based methods with IT architecturebased risk assessment and countermeasure definition and prioritization. Countermeasure prioritization is linked to business goals to achieve and based on cost of countermeasures and their effectiveness in reducing risks. RiskREP offers the potential to elicit complete security countermeasures, but also supports the deliberate decision and documentation of why the security analysis is focused on certain aspects. We illustrate RiskREP by an application to an action case.

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. How ever, 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) [9, 14]. We propose a method that allows 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.

Specification of misuse cases allows the specification of misuse threats [3, 6, 11, 13], but no method so far has added a link to business goals or countermeasure specification. We build on current proposals for extending RE methods with security risk assessment (RA) [2, 4, 8, 10, 12, 13]. (Interested readers may refer to [9] for an overview of the related work.) We aim at deliberate focusing of the analysis on the most important aspects and avoid applying inappropriate operations to scales like multiplications of likelihood with impact and subtractions. For example, if likelihood is estimated on an ordered, non-interval scale, then it makes no sense to add likelihoods or multiply with impact but can only do comparisons.

RiskREP is built on the CRAC++ [9] and MOQARE [5] methods developed by the authors. We present the metamodel of RiskREP in Section 2, present an action case in section 3, and discuss lessons learned in Section 4.

2 Meta model of RiskREP

The meta model (Figure 1) contains concepts from three perspectives, i.e. the *business perspective*, the *user perspective* and the *technical perspective*.

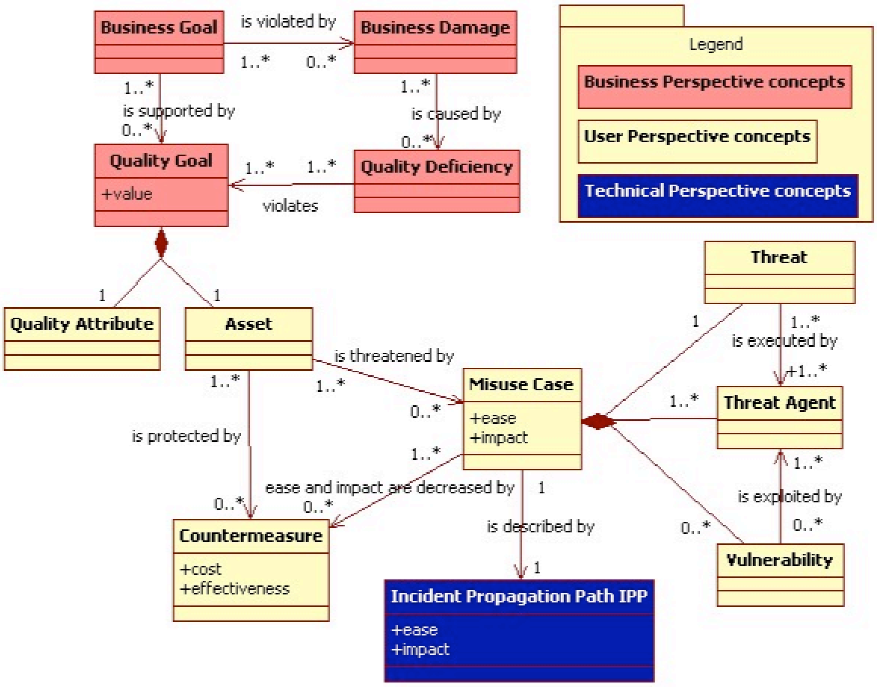

Figure 1: Meta-model showing the concepts and their interrelations.

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. They are non-functional system goals that support business goals. These goals are expressed as high-level quality requirements that consist of a quality attribute and an asset, like "confidentiality of password". They help to focus the analysis of the quality of an IT system on the most important quality attributes. 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 [1] 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 (MCs) [11] describe scenarios in which a threat agent can cause a quality deficiency. The MC 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 MCs 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 reduces the estimated impact at threat/vulnerability and/or the ease of threat execution. Countermeasures are expressed as quality 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 IPP.

Technical perspective: *Incident Propagation Paths* (IPPs) are descriptions of MC from the technical perspective. In some cases, an IPP 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 IPP scenarios are important for humans to imagine the flow of events including the causes and consequences of incidents. Like the MCs, the IPPs are prioritized based on their execution ease and the impact they have. There may be several IPPs realizing the same MC. The execution ease of a MC is an estimation of the effort required to carry out a MC. This effort is determined by the most resistant vulnerability that needs to be exploited to carry out the MC. 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 MC.

3 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 MC (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 countermeasure in reducing MC risk, and their cost.

At each of these steps, it is possible to either analyse the complete system, all business goals, all MC, 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 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. BG5, "gaining user acceptance", is threatened by one business damage, BD6, "Portal will not be used". 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 QD8 further. QD8 can be avoided by three high level quality goals, i.e. QG5: Confidentiality of assets, QG6: Integrity of assets, and QG7: Availability of assets.

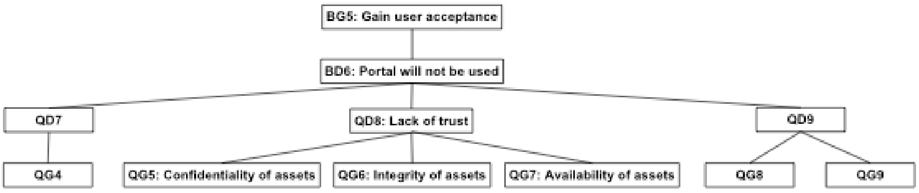
BG5: Gain user acceptance
BD6: Portal will not be used
QD7
QD8: Lack of trust
QD9
QG4
QG5: Confidentiality of assets
QG6: Integrity of assets
QG7: Availability of assets
QG8
QG9

Figure 2: Business concepts elicites with RiskREP

Step 2: Risk analysis

The risk expert first identifies possible misuse cases (MC) 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 (IPP) through the architecture that connects entry points of the system to the MC. This allow us the estimation of the ease of execution of the MC. Modeling the execution ease is also the main difference between IPPs and Misuse Case Maps [7].

The risk expert also assesses the value of each quality goal, for example by using value models for availability [14] or confidentiality [9] and then estimates the impact or damage caused by the MC to these quality goals.

For example, in the case study, MC5 (Manipulation of account data) threatens 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 assetsrelated to 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. (Due to space restriction we cannot provide the description of the scales here.) The execution ease of MC5 was estimated 1.5 and its impact was estimated 1. IPPs are described by the MC here and therefore we did not draw them. In total, related to QG6, we identified ten MCs, one of which we show in Table 1. As this table illustrates, the risk of a MC 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 MCs according to their risk.

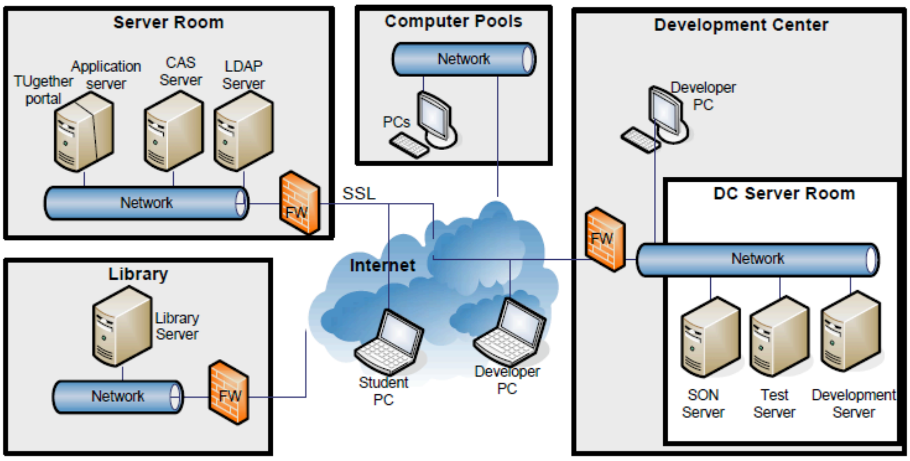

Figure 2: TUgether portals IT architecture. (FW: Firewall, DC: Data Center, CAS: Central Authentication Service, SON: Personal Development Server.)

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 threatvulnerability 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 1: Some MCs 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

To prioritize countermeasures, their effectiveness in reducing the risk of MC must be estimated. We define the impact of a countermeasure on a risk to be 0 if it neither affects the impact nor the execution ease of an MC; 1 if it decreases either impact or execution ease; and 2 if it decreases both.

Countermeasures interact with each other. For instance, some may be overlapping, or diminish each others 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.

We now prioritize countermeasures according to their cost and effectiveness. Just as for risk, no multiplications or additions can be done because the scales we use are ordered, but neither interval nor ratio scales. The security objectives of companies and their security strategies differ from each other. Therefore, RiskREP recommends to define a company-specific heuristic for the countermeasure prioritization. In this case study, we used categories of MC risks and countermeasures added values. For instance, the MC category "frequent, but harmless" describes MC where ease is high, but impact is low, and the the countermeasure category "low hanging fruit" contains countermeasures where cost is 0, execution ease is reduced or impact is reduced or both are reduced. We discussed the partial ordering of countermeasures according to their cost and effectiveness with the stakeholders to reach an agreed prioritization. The partial orderings or MCs by risk and of countermeasures by cost and effectiveness provided the stakeholders with structured arguments for choosing a set of countermeasures to implement

4 Analysis and discussion

Our action case study showed that RiskREP can be used and leads to a list of MC 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 current way of working. While RiskREP potentially could elicit all security requirements completely, at each step it is possible to focus on the most relevant aspects, e.g. most important quality goals, most important MC etc and to document this descision. 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 MC (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 IPP. 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 metamodel 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. 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.
2. 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.
3. G. Elahi and E. Yu. Modeling and analysis of security trade-offs - A goal oriented approach. Data Knowledge Engineering, 68:579–598, 2009.
4. 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.
5. A. Herrmann and B. Paech. MOQARE: misuse-oriented quality requirements engineering. Requir. Eng., 13(1):73–86, 2008.
6. 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.
7. 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.
8. 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.
9. A. Morali. IT Architecture-Based Confidentiality Risk Assessment in Networks of Organizations. PhD thesis, University of Twente, Enschede, The Netherlands, 2011.
10. J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E. Yu. Exploring alternatives during requirements analysis. IEEE Software, 18:92–96, 2001.
11. G. Sindre and A. Opdahl. Eliciting security requirements with misuse cases. Requir. Eng., 10(1):34–44, 2005.
12. D. Stamatis. Failure mode and effect analysis FMEA from theory to execution. American Society for Quality Press, 2003.
13. 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.
14. E. Zambon. Towards Optimal IT Availability Planning: Methods and Tools. PhD thesis, University of Twente, Enschede, The Netherlands, 2011.