SEVENTH FRAMEWORK PROGRAMME

RASEN

Compositional Risk Assessment and Security Testing of Networked Systems

Deliverable D3.1.1

Baseline for Compositional Test-Based Security Risk Assessment

Project title:	RASEN
Project number:	316853
Call identifier:	FP7-ICT-2011-8
Objective:	ICT-8-1.4 Trustworthy ICT
Funding scheme:	STREP – Small or medium scale focused research project

Work package:	WP3
Deliverable number:	D3.1.1
Nature of deliverable:	Report
Dissemination level:	PU
Internal version number:	1.0
Contractual delivery date:	2013-01-31
Actual delivery date:	2013-01-31
Responsible partner:	Fraunhofer

Contributors

Editor(s)	Jürgen Großmann (Fraunhofer), Johannes Viehmann (Fraunhofer)
Contributor(s)	Jürgen Großmann (Fraunhofer), Bruno Legeard (SMA), Fabien Peureux (SMA), Fredrik Seehusen (SINTEF), Bjørnar Solhaug (SINTEF), Johannes Viehmann (Fraunhofer), Frank Werner (SAG)
Quality assuror(s)	Erlend Eilertsen (Evry), Tobias Mahler (UiO)

Version history

Version	Date	Description
0.1	12-11-05	Draft TOC
0.2	12-12-04	First draft of Section 2
0.3	12-12-13	First draft of Section 3
0.4	13-01-03	Merge of content from SMA, FOKUS, SINTEF, and SAG
0.5	13-01-11	First complete version
0.5	13-01-16	Finalization for internal review and quality assurance
1.0	13-01-31	Final version

Abstract

The overall objective of RASEN WP3 is to develop tools and techniques for compositional and test-based security risk assessment. This deliverable presents the industrial and scientific state of the art related to this objective. The presentation is structured according to the main research tasks of WP3, which are to develop support for i) compositional security risk assessment, ii) test-based risk identification and estimation, and iii) continuous risk assessment of large scale systems by use of test-based indicators. Considering the state of the art and the research objectives of WP3, we moreover identify the baseline for the WP3 research activities. The baseline is the tools and techniques that may serve as promising starting points for further development. The identification of the baseline was guided by the relevant research questions that are addressed by the RASEN project.

Keywords

Risk management, security risk assessment, security testing, risk composition, test-based risk analysis, continuous risk analysis, state of the art

Executive Summary

The overall objective of RASEN WP3 is to develop tools and techniques for compositional and test-based security risk assessment. Moreover, these tools and techniques will be leveraged to develop adequate and efficient techniques for continuous risk assessment.

The purpose of this document is twofold. First, we give an overview of the state of the art that is relevant for the WP3 research objective. Second, we present the RASEN WP3 baseline which is the existing tools and techniques that serve as a promising starting point for the research tasks.

The description of the state of the art and the baseline are organized according to the three main research tasks of WP3, which are to develop tools and techniques to support

- T3.1: Compositional security risk assessment
- T3.2: Test-based risk identification and estimation
- T3.3: Continuous risk assessment of large scale systems by used of test-based indicators

The discussion of the state of the art and the identification of the WP3 baseline are guided by the RASEN research questions that are relevant for this work package. These research questions are the following:

1. To what extent can we use composition to make security assessment of large scale networked systems more feasible and understandable?
2. To what extent can we relate criteria for valid composition of security risk assessment results to criteria for valid composition of security test results?
3. To what extent can we support reuse of security risk assessments to make security assessment of large scale networked systems more feasible?
4. What are good methods and tools for aggregating test results—obtained by both active testing and passive testing (monitoring)—to the risk assessment?
5. How can test results be exploited to obtain a more correct risk picture?

Considering the challenges of managing security and assessing the risks of the complex, heterogeneous and dynamic networked ICT systems and services of today, these are timely research questions. As shown in this deliverable, there exists little or no adequate support for compositional risk assessment and test-based risk assessment. While using existing techniques for risk modeling and test-based security risk assessment that provide some basic support, WP3 will develop novel techniques for modular risk modeling that can be combined with security testing. In turn, this approach to compositional and test-based risk assessment will serve as a basis for the RASEN techniques for reuse of risk assessments to support continuous security risk assessment.

Table of contents

1 Introduction

Organizations, enterprises, people, governments and the European society as a whole increasingly depend on ICT systems. The criticality of ICT systems requires adequate mechanisms, methods and tools for managing security and maintaining security risks at an acceptable level for all stakeholders. This is particularly the case for critical infrastructures such as telecommunication, public health, banking and power supply. Moreover, trustworthy and secure ICT systems are required for many applications in several market sectors, including eCommerce, eGovernment, and eHealth.

At the same time, such systems and infrastructures become more and more heterogeneous and complex. Moreover, the networked service and computing environments increasingly cross organizational and geographical borders, further contributing to increase the security challenges. Assessing the security risks of such complex systems in their entirety is infeasible, and therefore requires means for assessing individual parts or aspects of the system in isolation and subsequently combining the results for understanding the overall security risks. Such an approach will also facilitate continuous security risk assessment of evolving systems, since it allows only the parts of the system that have changed to be reassessed.

A further challenge of security management is how to combine enterprise-level security assessment with the security assessment of the underlying technologies. The former is supported by high-level means such as security risk assessment, whereas the latter is supported by low-level means such as security testing. Appropriate means for combining these levels should facilitate test-based security risk assessment and risk-based security testing.

The objective of RASEN WP3 is to develop tools and techniques for compositional security risk assessment supported by security testing. This includes tools and techniques for modular risk assessment where risk models can be composed and decomposed; tools and techniques for identifying, estimating and verifying security risks based on security test results; and tools and techniques for reuse of risk assessment and security test results, as well as dynamic updates of risk assessment results based on test results.

This deliverable gives an overview of the state of the art within compositional security risk assessment and test-based security risk assessment. The purpose of the deliverable is moreover to establish the baseline for the R&D work of RASEN WP3. The baseline is the existing techniques and tools that are promising and may serve as a basis for the upcoming project work within WP3. The description of the state of the art and the identification of the baseline are guided by the relevant research questions that need to be addressed when tackling the WP3 research challenges. These, extracted from the RASEN DoW, include the following:

- To what extent can we use composition to make security assessment of large scale networked systems more feasible and understandable?
- To what extent can we relate criteria for valid composition of security risk assessment results to criteria for valid composition of security test results?
- To what extent can we support reuse of security risk assessments to make security assessment of large scale networked systems more feasible?
- What are good methods and tools for aggregating test results—obtained by both active testing and passive testing (monitoring)—to the risk assessment?
- How can test results be exploited to obtain a more correct risk picture?

This document is structured according to the three main research tasks of WP3 as described in the RASEN DoW. Hence, in Section 2 we give an overview of the state of the art within compositional security risk assessment, addressing the R&D challenges of task T3.1. In Section 0 we present the state of the art within test-based risk identification and estimation, addressing the challenges of task T3.2. In Section 0 we present the state of the art on continuous risk assessment, which is related to research task T3.3. Based on the state of the art and the RASEN research questions, we present the WP3 baseline in Section 0. Finally, we conclude by summarizing in Section 0.

2 Compositional and Modular Security Risk Assessment

Risk assessment involves the identification, analysis and evaluation of security risks with respect to identified assets [19]. Security risk assessment focuses on information assets and security properties such as availability, confidentiality and integrity [21][23]. Other relevant security properties to consider include authentication, non-repudiation and authorization [12]. Risk identification is the process of finding, recognizing and describing risk; risk analysis is the process of comprehending the nature of risk and to determine the level of risk; risk evaluation is the process of comparing the results of risk analysis with the risk criteria to determine whether the risk and/or its magnitude is acceptable or tolerable [20].

In this section we give an overview of risk assessment techniques and the state of the art for compositional and modular risk assessment. A risk assessment technique is a means to determine which incidents can happen and why, what the consequences are, what the likelihoods of their occurrence are, and whether there are any factors that may mitigate the likelihood or consequence of incidents [20].

The section is structured as follows. In Section 2.1 we present the principles of modularity and motivate a compositional and modular approach to risk assessment. In Section 2.2 we give an overview of existing security risk analysis techniques, and in Section 2.3 we present the state of the art on techniques for modular assessment. In Section 2.4 we address risk management in the organizational context and at enterprise level.

2.1 Benefits of Modularity

By modular risk assessment we mean a process for assessing separate parts of a system or several systems independently, with means for combining separate analysis results into an overall result for the whole system [5]. Modularity is facilitated by decomposition, which is a well-known feature from system specification and design [37]. Decomposition is the process of partitioning a system specification into separate modules that can be developed and analyzed independently, thus breaking the development problem into more manageable pieces. Each module may moreover be developed at different sites, by independent teams, or within different companies [4].

There are several benefits of a modular and compositional approach to risk assessment. First, if different risk assessments have already been conducted for different parts of a system, composition allows the separate results to be combined into a more global risk picture. Second, for large, complex systems, modularity allows the system to be decomposed into a number of sub-systems the risks of which are assessed separately. Third, if a software component or application is reused in different systems, a separate risk assessment can be conducted for this component or application alone, instead of assessing it from scratch in the different contexts of use. Fourth, if a system changes and evolves, modularity allows the whole system to be reassessed by assessing only the parts that have changed.

2.2 Risk Assessment Techniques

A comprehensive overview of risk assessment techniques is given in the ISO/IEC 31010 standard [24]. The techniques are classified by which steps of the risk assessment process they support, i.e. identification, analysis and/or evaluation. For risk analysis, the techniques are further classified according to the kind of analysis that is supported, mainly consequence analysis and/or likelihood analysis. The techniques may furthermore be qualitative, semi-quantitative or quantitative.

Brainstorming is a common risk assessment technique, and is a means of collecting a broad set of ideas from different experts and sources. The technique may be stimulated by prompts or interview techniques. The Delphi technique [26] and HazOp (Hazard and Operability) [18] analysis are examples of brainstorming techniques to extract and combine expert opinions in a structured way. Similarly, SWOT [14] and FMEA/FMECA [3] are based on brainstorming and typically uses table formats to document the results. SWOT is used to evaluate the strengths, weaknesses, opportunities and threats associated with a project or business activity. In security risk assessment it is used to get an initial overview of the risk picture and helps scoping the analysis to focus it on where it is most needed. FMEA/FMECA (Failure Mode Effect Analysis / Failure Mode Effect and Criticality Analysis) is a method to assess the potential failures of individual components within a system. The method is usually conducted in two steps. First, FMEA is used to identify failure modes and their effect. Second, FMECA is used to conduct a criticality analysis to determine the significance of each failure mode, qualitatively or quantitatively. Although such brainstorming techniques can be used to support risk assessment in modular approaches, they do not come with means or techniques for modeling risks or for how to reason about combined results.

Risk modeling is a technique to aid the process of identifying and estimating likelihood and consequence values. A risk model is a structured way of representing an incident with its causes and consequences by means of graphs, trees or block diagrams [32].

Markov models [13][17] describe systems by a set of operations and failure states, and the transitions between these states. The models may also include the description of barriers that should prevent an attack or reduce the consequences of an attack. Transitions may be annotated with probabilities in order to enable Markov analyses and reasoning about the probabilities.

Bayesian networks [2] are directed acyclic graphs where the nodes represent causes or contributing factors with respect to related nodes. A Bayesian network is a graphical, probabilistic model that can be used both qualitatively to represent relations between causes and effects, and quantitatively to compute probabilities.

Fault tree analysis (FTA) [16] uses fault trees to describe and analyze the causes of an event or incident by breaking it down into smaller fault events. The events are structured into a binary tree with logical gates (e.g. and/or) that shows conditional dependencies between the faults. FTA supports documentation and reasoning about probabilities using well-defined rules.

As described in [44], for calculating probability values in FTA, the following basic equations/formulas can be used. If $\mathrm{P}(Y)$ is the probability that some incident Y occurs and if $\mathrm{P}(X|\ Y)$ is the conditional probability for incident X given that it is known that incident Y occurs, then the probability $\mathrm{P}(X\cap\ Y)$ that both incidents X and Y occur can be calculated with $\mathrm{P}(Y)$ and $\mathrm{P}(X|\ Y)$:

$$\mathrm{P}(X \cap Y) = \mathrm{P}(X|Y) * \mathrm{P}(Y) \tag{1}$$

If $\mathrm{P}(X)$ is the probability that X occurs, then the probability $\mathrm{P}(X \cup Y)$ that at least one of the two incidents X, Y occurs is:

$$\mathrm{P}(X \cup Y) = \mathrm{P}(X) + \mathrm{P}(Y) - \mathrm{P}(X \cap Y) \tag{2}$$

If X and Y are statistically independent (i.e. $\mathrm{P}(X|\ Y) = \mathrm{P}(X)$ and $\mathrm{P}(Y|\ X) = \mathrm{P}(Y)$), then:

$$P(X \cap Y) = P(X) * P(Y) \tag{3}$$

$$\mathrm{P}(X \cup Y) = \mathrm{P}(X) + \mathrm{P}(Y) - P(X) * P(Y) \tag{4}$$

If $\mathrm{P}(X|\ Y) = 1$ and $\mathrm{P}(Y|\ X) = 1$, then:

$$P(X \cap Y) = P(X \cup Y) = P(X) = P(Y) \tag{5}$$

The probability value *V* for an incident that has to be triggered by at least *threshold* Ψ of *n* statistically independent incidents each having the probability *p* can be calculated using the following binomial formula [45]:

$$\mathrm{V} = \sum_{k=\Psi}^{n} \binom{n}{k} * p^k * (1-p)^{n-k} \tag{6}$$

Calculating probability values is a vital part of risk analysis since it provides a basis for the estimation of risk levels. Multiple algorithms are known for calculations in FTA, including the binary decision diagram (BDD) based algorithm presented in [43] and DIFtree [41] using both BDD and Markov chains. Various software tools support FTA, e.g. Galileo [40], and these tools can, for example, calculate the top level incident's probability values taking just the base incidents probability values as input.

A variant of fault trees specific to the security domain is attack trees [33]. An attack tree has an attack goal as its top node, and the branches of the tree explore different ways of reaching this goal by an attacker.

Event tree analysis (ETA) [15] is similar to FTA but works in the opposite direction. The technique uses event trees to start with the incident and explore its possible consequences. The exploration takes the form of a binary tree where each branching point is a success/failure of a defense mechanism. The probability of each consequence can be calculated by aggregating the probabilities of the defense successes and failures that lead to the consequence.

Cause-consequence diagrams [29][32] combine features of fault trees and event trees. Starting with the incident, a cause-consequence diagram is constructed backwards to identify its causes (fault tree), and forward to identify its consequences (event tree). Also the Bow tie technique [34] can be considered to be a combination of fault trees and event trees by its support for identifying and documenting both the causes and consequences of an incident.

It can be beneficial to do both, FMEA/FMECA and FTA because they have complementary strengths. In [39], a combination of both is suggested as "Bouncing Failure Analysis (BFA): The Unified FTA-FMEA Methodology".

Microsoft has developed their own technique called Threat Modeling [35], which uses Data flow diagrams (DFDs) to graphically represent the target of analysis. Based on such diagrams, the target is decomposed into components that are assessed with respect to identified threats. The technique uses threat trees for the modeling and analysis of attack paths, and uses the STRIDE [12] checklist approach to identify risks with respect to security properties. The OCTAVE method for security assessment [1] uses its own tree notation. The language has much in common with event trees and fault trees by the support for modeling both the causes for and the outcomes of unwanted incidents.

CORAS [28] is a model-driven approach to risk assessment that comes with its own notation for risk modeling by means of so-called threat diagrams. A threat diagram is an acyclic graph documenting how threats exploit vulnerabilities to initiate threat scenarios that may lead to unwanted incidents. The diagrams can be annotated with likelihoods and consequences, and are supported by techniques for reasoning about likelihoods. These techniques include rules for probability reasoning that are similar to the rules presented above in the context of FTA.

Risk graphs [5] are a modeling technique to aid the identification and structuring of scenarios and events leading to unwanted incidents, and to facilitate the estimation and reasoning about their likelihoods. Risk graphs come with a formal syntax and semantics, and a calculus is provided to support formal and rigorous reasoning about the models. An advantage of risk graphs is that they can be understood as a common abstraction of several established risk modeling techniques, such as fault trees, event trees, cause-consequence diagrams, Bayesian networks and CORAS threat diagrams. This means that the calculus and analysis techniques developed for risk graphs can be instantiated by these notations, and therefore carry over to them. In [5] this is demonstrated and exemplified for CORAS threat diagrams.

The risk assessment techniques reported so far in this section assume that the target of analysis is well-understood. The target of analysis is the organization, system, services, processes, etc. the risks with respect to which need to be understood and documented. For large, complex systems that are too complex to be analyzed in detail as a whole, the analysts may split the target to more manageable sub-systems that are analyzed separately. However, none of the aforementioned techniques come with specific support for how to compose the different results and methodically derive the global risk picture. A main objective of RASEN is to develop such support while leveraging existing techniques for risk modeling and risk assessment such as those presented in this section. Further techniques to consider as part of the RASEN baseline are the techniques for modular risk assessment discussed next.

2.3 Techniques for Modular Risk Assessment

Although modularity and decomposition are recognized and well-established techniques within system development and design, such techniques are almost non-existent within risk management and security risk assessment. As observed in [36], one of the problems is that few traditional risk assessment methods take into account that the risk level towards component-based systems may change given changes in the environment of the systems. Existing risk assessment methods and techniques are largely monolithic in the sense that systems in general are analyzed as a whole [27] and lack means for deducing the effect of composition with respect to risk. In the following we present the techniques we are aware of that provide some support for modular and compositional risk assessment.

Some approaches to hazard analysis address the propagation of failures in component-based systems. Several of these approaches describe failure propagation by matching ingoing and outgoing failures of individual components. In [9][10] UML [30] component diagrams and deployment diagrams support a method for compositional hazard analysis. Fault trees are used to describe hazards and the combination of component failures that can cause them. For each component, the method is used to describe a set of incoming failures, outgoing failures, local failures (events) and the dependencies between the former two. Failure information of components can be composed by combining their failure dependencies.

A version of FMEA [3] is applied in [31] to support a technique that focuses on component interfaces. The technique is used to describe the causes of output failures as logical combinations of internal component malfunctions or deviations of the component inputs. Propagation of faults in a system is described by synthesizing fault trees from the individual component results.

A method for compositional fault tree analysis is proposed in [25]. Component failures are described by specialized component fault trees that can be combined into system fault trees via input and output ports.

The work presented in [8] addresses the problem of predicting risks related to introducing a new component into a system. The approach applies Bayesian networks to analyze failure probabilities of components. Quantitative and qualitative evidence concerning the reliability of a component is combined, and Bayesian networks are used to calculate the overall failure probability.

The model-driven performance risk analysis method in [7] takes into account both system level behavior and hardware specific information. The method combines performance related information of interaction specifications with hardware characteristics in order to estimate the overall probability of performance failures. The approach is based on a method for architectural-level risk analysis using UML [11].

In [5] dependent risk graphs are introduced as a technique to support modular risk modeling and assessment. Dependent risk graphs extend the risk graph notation with support for documenting and reasoning about assumptions and dependencies. The approach uses an assumption-guarantee style by the division of risk graphs into two parts, namely the assumption and the target. The assumption documents the assumptions on which the risk analysis depends and the target documents risk analysis results of the target of analysis. Typically, the assumptions are about the environment of the target.

Dependent risk graphs facilitate modular risk assessment by the support for decomposing the target of analysis and later combining the assessment results. For example, when decomposing a target system into two, the target in one may serve as the assumptions in the other and vice versa. Once the two separate risk assessments are completed, a calculus provides rules for how to combine the results into one risk graph. The calculus characterizes the conditions under which the assessment of complex scenarios can be decomposed into separate assessments that can be carried out independently, and the conditions under which risk assessments of separate system parts can be put together to provide a risk analysis for the system as a whole.

As mentioned above, by using risk graphs as the basis the technique can be instantiated in other established risk modeling approaches. In [5] this is demonstrated by the instantiation in CORAS. In [6] this modular and component-based approach to risk assessment using CORAS is integrated into a component-based system development process to support risk assessment in the development process. In [28] the instantiation in CORAS is further elaborated, resulting in an extension referred to as Dependent CORAS.

In [46] an extension of CORAS is suggested that explicitly supports components by representing them with reusable *threat interfaces*. *Threat interfaces* have vulnerabilities as input ports and unwanted incidents as output ports. Modeling the relations between the output and input ports of *threat interfaces* generated for individual components in a *threat composition diagram*, the probability values of unwanted incidents for the complex system composed of these components can be calculated.

This concept is similar to ETA and FTA. With gates and dependency sets it can express relations that conventional CORAS diagrams cannot model well. But in contrast to a tree, there can be multiple top level incidents. Incidents can have relations leading to more than a single vulnerability and to several new incidents. Therefore, dependencies can be modeled directly as common trigger nodes. In a fault tree, a fault triggering *n* other faults must be represented by *n* nodes having the same name, but no graphical connection, which is less intuitive. Even more important, bouncing analysis becomes feasible by going top-down and bottom-up within the same model. Using FTA, bouncing analysis is only possible in combination with other risk analysis methods like ETA or FMECA, which work on other models than fault trees. However, transitions between different methods can cause problems and might be too difficult to be feasible in practice.

Conventional risk diagrams can be created directly from the *threat composition diagram*, which is the next step in the CORAS risk analysis method. Nevertheless, it might be helpful to keep the component information for further analysis. The idea is to make components or complex combinations of components comparable in terms of risks. Typically, for a complex system, there is not only one single possible configuration. The system could probably be build using another combination of components or using completely other base components, too. It should be possible to choose the architecture with the fewest risks and to communicate such a design decision. Therefore, the component based risk comparison diagram is introduced in [46], which can be derived from the threat composition diagram.

2.4 Enterprise-Level GRC Frameworks

Governance, risk management and compliance (GRC) typically covers an organization's activities regarding corporate governance, enterprise risk management, and compliance with governmental laws and regulations [79]. In recent years the maturity of GRC analytics, frameworks and tool-sets has increased and thereby contributed to reducing the decision making timelines of organizations.

One such framework is the ARIS [80] GRC tool-set. As depicted in Figure 1, the solution consists of many software components that integrate into the GRC [79] architecture. The most essential module to use in terms of security risk assessment is the *Operational Risk Management* component which operates on real-time data and ensures compliance and workflows. In addition the *Modeling and Process Risk Simulation* [76] module evaluates aspects such as changes in processes or installation of controls to mitigate risks. The use of dash-boarding techniques allows a real-time illustration. The latter module can be used as a starting point when implementing a GRC solution. Is offers support for process modeling, as well as the definition of risks and controls, along with the related responsibilities. By the use of the central ARIS repository [78] no copies are created, but a single point maintained which integrates all GRC related data. For detailed models, user-specific risk structures in combination with the Bow tie methodology [34][77] can be used as modeling means. Hereby simulations assure the correctness and compliance of models with the user’s intention by conducting what-if analyses of occurring risk and offering means to define test definitions and risk assessments.

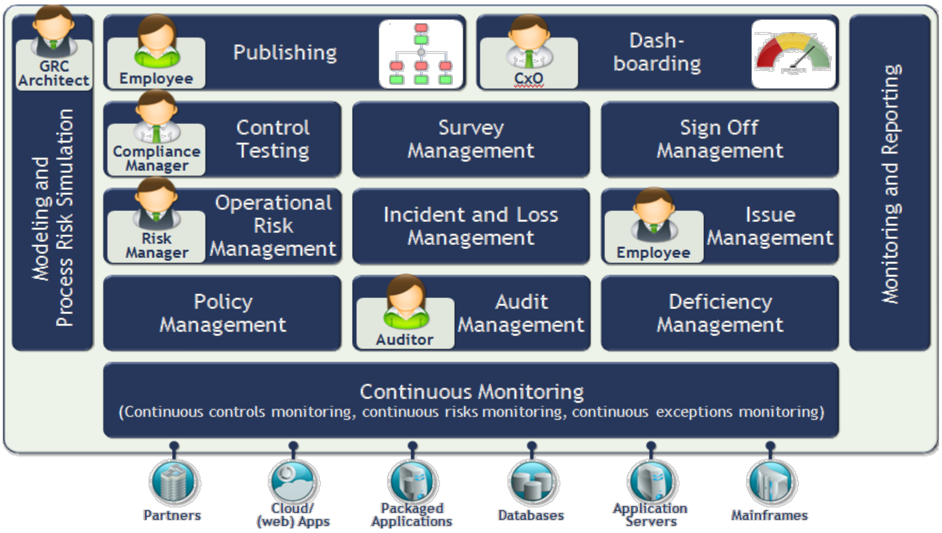

Figure 1 – Architecture of Software AG GRC approach

Generally, the modeling that is conducted using this framework unifies different views to combine the business process management (BPM) view and the GRC view. Showing and documenting regulatory requirements, risks, business processes and business organizations in one model increases transparency and decision support, and also facilitates the identification and documentation of dependencies between the different views. A further interesting module is the *Continuous Monitoring* module for real-time assessment and continuous risk monitoring. This module is presented in more details in Section 0 below.

The objective of RASEN PW3 is to develop techniques and tools for compositional security risk assessment that can be integrated into a GRC tool-set such as the ARIS framework. Organizations and enterprises should be able to make effective use of their existing frameworks while leveraging the RASEN results. Currently, industrial tools such as ARIS do not have efficient mechanisms for compositional risk management, although dependencies between the different views are an aid to identify the parts of the enterprise that may need to be reassessed. In addition to risk monitoring, such enterprise frameworks should also benefit from techniques for test-based security risk assessment, which are discussed in the next section.

3 Test-Based Risk Identification and Estimation

A risk assessment is often performed at a relatively high level of abstraction and is often heavily dependent of expert judgment. This can result in a large degree of uncertainty both about the existence of faults or vulnerabilities in the target of analysis and the estimated likelihoods of risks. For this reason, it is beneficial to combine the risk assessment with other lower-level more technical assessment methods in order to reduce uncertainty and to uncover vulnerabilities or faults that are difficult to identify at the risk assessment level. One way of doing this is to combine risk assessment with testing, using the test results to verify and/or update the risk picture.

In this section we give an overview of techniques that use testing for improving/verifying the risk assessment. In Section 3.1 we motivate the use of test-based security risk assessment by summarizing relevant guidelines from international standards on information security. In Section 3.2 we give a summary of relevant state of the art approaches and highlight their main strengths and weaknesses.

3.1 From Test Results to Security Risk Assessment

The risk management process as defined by the ISO 31000 standard consists of the activities depicted in Figure 2. In the ISO/IEC 27005 standard [23], this process is adapted to information security risk management. As explained in the latter standard, "*this information security risk management process can be iterative for risk assessment and/or risk treatment activities. An iterative approach to conducting risk assessment can increase depth and detail of the assessment at each iteration. The iterative approach provides a good balance between minimizing the time and effort spent in identifying controls, while still ensuring that high risks are appropriately assessed*".

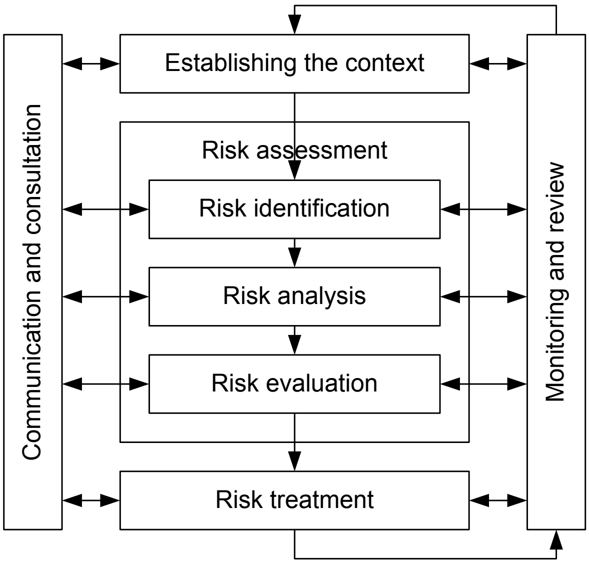

Figure 2 - Risk management process

An important issue during the risk assessment phase centers around the evaluation of uncertainty [38]. After identifying possible incidents, it is necessary to assess the likelihood of each incident and its impact when occurring, using qualitative or quantitative analysis techniques [23]. This should take account of how often the threats occur and how easily the vulnerabilities may be exploited. This is where vulnerability testing on the ICT system plays a critical role, particularly for deliberate threat sources. The possible exploitation of vulnerabilities on the ICT systems should be considered carefully on the results of vulnerability test execution; particularly for ICT systems open on the internet (such as Web applications).

In ISO 27005, vulnerability testing is explicitly defined as a key means to assess the vulnerabilities of ICT systems: "*Proactive methods such as information system testing can be used to identify* *vulnerabilities depending on the criticality of the Information and Communications Technology (ICT) system and available resources (e.g. allocated funds, available technology, persons with the expertise to conduct the test)*".

"Security testing and evaluation is another technique that can be used in identifying ICT system vulnerabilities during the risk assessment process. It includes the development and execution of a test plan (e.g. test script, test procedures, and expected test results). The purpose of system security testing is to test the effectiveness of the security controls of an ICT system as they have been applied in an operational environment. The objective is to ensure that the applied controls meet the approved security specification for the software and hardware and implement the organization's security policy or meet industry standards" [23].

Therefore, ISO 27005 establishes clearly the security testing techniques, such as code review, static and dynamic application security testing, monitoring and web application vulnerability scanning, should play a key role in the risk assessment process to evaluate the potentiality of identified vulnerabilities. The standard makes a clear link between security testing and risk assessment, highlighting that test results analysis should be used in the risk evaluation phase regarding information security properties. However, the standard does not provide any process, method or guideline on how to systematically exploit results of security testing in risk assessment. In the next sub-section we give an overview of existing techniques that provide some support of that kind.

3.2 Techniques for Test-Based Security Risk Assessment

There are quite a number of approaches that use risk assessment to improve the testing process. However, there are not so many approaches that go in the other direction, i.e. that use testing to improve the risk assessment in a systematic and methodic way. The only relevant approaches that we are aware of are Benet [81], Wong et al. [84], and Erdogan et al. [82]. In the following, we will give an overview of these.

The main purpose of the approach given by Benet [81] is to verify that software risk mitigations are put into place and that they are actually effective. It starts by identifying the intended usage of the system and their related user requirements. This is further used, together with other system documentation and customer feedback, as a basis for risk identification. The identified risks are then assessed and a set of risk mitigations are identified for the risks that are unacceptable in form of risk-related requirements. Finally, testing is used as a means to verify that the risk mitigations are put into place and that they are effective. Thus, the approach uses testing to verify and validate the risk picture obtained from the risk assessment.

The approach also provides a clear traceability between the risk-related requirements and the test cases that eventually verify the risk mitigations. The traceability is specified by means of a traceability matrix. Furthermore, the approach is mainly concerned about safety risk and gives guidelines for what to consider when designing the test cases with respect to safety. The traceability matrix makes it clear how the test results impact the risk assessment. If a risk related test case fails, this means that a risk-related requirement is not satisfied, and that its mitigating effect can be ignored. Conversely, if all risk related test cases pass, confidence has been gained that all the risk-related requirements are satisfied, and that their mitigating effects are in place.

The approach mainly focuses on verifying risk mitigations and providing a clear traceability between risk-related requirements and the test cases. It does not elaborate on how risk assessment is carried out or how testing is carried out (except the guidelines for what to consider when designing the test cases with respect to safety). However, risk assessment is supported by a table based risk assessment technique in which one table is created per risk related requirement. The table consists of a description of the risk related requirement, the risk, the risk level, the risk mitigation and identifiers for the test cases that have the objective to verify the risk mitigation.

The approach is motivated by explaining its application on a real project concerning an in vitro diagnosis (IVD) instrument. It is not supported by any dedicated tool and does not specifically address security, but mentions the importance of testing for security in relation to safety.

From the perspective of using testing to improve the risk assessment, one of the main strengths of Benet's approach is that the impact of the test results on the risk picture can be calculated automatically. Tests are derived from requirements to mitigation mechanisms, and the mitigating effect of a requirement on a risk is clearly specified. A test result has one of two values, namely pass or fail. If the test fails, we can assume that the mitigation effect of the requirement tested is not in place. In that case the overall risk level can be calculated by ignoring that mitigation effect.

A weakness of the approach is that it is intended to be used as part of a software development process, where the aim is to test mitigation mechanisms that have not already been developed. This limits the applicability of the method since in many cases it is also of interest to analyze systems or mechanisms that have already been developed. Another weakness of the paper itself is that it lacks details about the process and the techniques that are used; the paper reads more like a discussion of an idea than a presentation of process or a technique. Moreover, only the failure of tests has an impact on the risk picture, i.e., if all the tests pass, the risk picture remains unmodified. Finally, the paper only addresses safety and not negative testing, i.e. testing on invalid input or unexpected user behavior.

Wong et al. [84] present an approach to the identification of so-called risk of code. Risk of code is described as the likelihood that a given function or block within source code contains a fault. Thus, what the authors really present is an approach to the identification of the likelihood that a given function or block within source code contains a fault. A fault within source code may lead to execution failures, for example abnormal behavior or incorrect output.

The approach proposes two kinds of risk models (static risk model and dynamic risk model) for the purpose of calculating the risk of code in terms of likelihood, as described above. The risk models are explained in detail in a case study. The approach does not specifically address security.

The static risk model and the dynamic risk model are used to calculate the risk of code by using five different metrics: Number of variable definitions (V), number of function calls (F), number of decisions (D), number of c-uses (C), and number of p-uses (P). The main difference between the static and the dynamic risk model is that the latter uses test coverage to calibrate the actual value of each metric component. It is up to the user of this approach to select which metrics to include in the risk model. Furthermore, each metric is multiplied with a weight factor for the purpose of giving either more or less emphasis to the metrics.

The identification of the risk of code is supported by a dedicated tool and is carried out in three main steps. The first step is to generate a graph of the functions and their interactions in order to capture their connectivity. The functions are represented as nodes while their interactions are represented as edges. The graph also has a unique start node and end node representing the functions at which execution begins and ends, respectively. The second step is to identify test cases; each path that starts from the start node and ends at the end node is identified as a test case. The third step is test execution and identifying the risk of code by making use of the test results as an input to the risk model.

From the perspective of test-based risk assessment, the most interesting part of the approach is the technique for updating the risk model based on test results. The idea is to calibrate/update the values of the code metric components, such as number of function calls, based on test coverage. For example, suppose we want to compute the risk of a source code block *b* and that the metric component F for block *b* has the value 20, meaning that the block has 20 potential function calls. As the tests are executed, the value F is reduced by each successful test that covers a potential function in *b* that has not yet been covered by any other test. For example, if 8 functions are covered, the value F will be reduced to 12. In this way, the more *b* is tested, the less its risk value will be.

The main strength of the approach is that many of the activities are automated and tool supported. Both the risk assessment step and the risk model update step can be performed automatically. In addition, the approach is more granular when it comes to updating the risks values based on test results than the approach of Benet [81]. The main weakness of the approach is that it is only applicable to source code analysis. In addition, only the likelihood of risks are taken into account (not consequences), and the approach does not consider security.

Erdogan et al. [82] describe the experiences of applying a method for test-based risk assessment in a case study. The method has three main generic phases: Phase 1 is to establish the context and target of evaluation, and to carry out a security risk analysis of the target of evaluation. The analysis relies mostly on expert judgment from the analysis participants. Phase 2 is to generate and execute security tests that explore the risks identified during the security risk analysis. Phase 3 is to validate and update the risk model (produced in phase 1) based on the security test results.

Each of these generic phases is decomposed into one or more steps. The output of phase 1 is a CORAS [28] risk model describing threats, vulnerabilities, threat scenarios and risks. In phase 2, threat scenarios are prioritized, and the ones that have the highest priorities are detailed into test cases that are executed. In phase 3, the CORAS risk model is updated based on the test results.

Erdogan et al. also describe the extent to which the risk model had to be updated based on the test results in the case study they performed. Their main conclusion is that new vulnerabilities were deleted/added from/to the risk model as a result of the testing, and that likelihood values in the risk model had to be updated as a result of this. However, the authors do not propose any technique or detailed guideline for how to update the risk model based on the test results.

One of the main strengths of the approach proposed in [82] is that it is based on precise risk models and that it is applicable to a wide range of systems (not only, for example, source code as in the case of Wong et. al. [84]). The advantages of having precise risk models as opposed to informal tables (as in the case of Benet [81]) are that it simplifies tool support, and it enables precise characterization of test prioritization and impact of test results on the risk model. The main weakness, as seen from a test-driven risk assessment perspective, is the lack of techniques for updating the risk model based on the test results. Using the approach, this has to be performed completely manually and based on expert judgment.

In Table 1 we have given a summary of the main strengths and weaknesses of the approaches to test-based security risk-assessment that are discussed in this section.

Approach	Strengths	Weaknesses
Benet [81]	Impact of test results on risk picture is easily calculated	Tests that pass have no impact on the risk picture (only failed tests taken into account)
Wong et al [84]	Impact of test results on risk picture is calculated automatically	Approach is only applicable to source code analysis
Erdogan et al [82]	Based on precise risk models	No automated or structured technique proposed for updating the risk picture based on test results

Table 1 – Summary of strengths and weaknesses of three approaches to test-based risk assessment

In the R&D activities of WP3 we will carefully explore the potential of leveraging the strengths of these three approaches when tackling the relevant RASEN research questions. While doing so we aim not only to better understand how to aggregate test results at the risk assessment level, and how to exploit test results to obtain a more correct risk picture. We also aim to make test-based security risk assessment to work in the setting of compositional risk assessment. For this challenge, there is to the best of our knowledge no support in the state of the art. In fact, and as discussed in this and the previous section, there exists little support for compositional risk assessment and test-based security risk assessment alone.

4 Continuous Security Risk Assessment

For systems that change and evolve, the associated risks evolve too and should be understood and assessed as such. Traditional methods for risk assessment offer little support for handling change in a systematic way in order to continuously maintain the validity of risk assessment results when changes occur [27]. A straightforward solution is to conduct a full risk assessment from scratch when faced with a potential risk relevant change. However, this is not only very time and resource consuming; it also often implies conducting exactly the same analysis again for all parts of the target system that are not affected by the changes.

Considering enterprises, companies and other organizations of today, these are constantly facing an increasing complexity of ICT systems and software infrastructures. Key drivers for this are certainly the efforts to deal with the vast amount of data which needs to be conquered, as well as the sheer complexity of modern appliances in software infrastructures. These issues can be partly mitigated by an increase in automation in business processes. Most of these changes in systems and software components are inherently related to an increasing complexity of the underlying business and ICT infrastructures in lieu. In addition, external requirements and customer needs must be taken into account, adding to the complexity of ICT processes and infrastructures by the need for assurance of compliance with these needs and changing external requirements. To cope with the speed of changes while retaining in focus the aspects of risk awareness, legal issues and the company’s strategic, the need for tools and techniques to support continuous security risk assessment increases.

In a similar way, software vendors are confronted with issues of growing software complexity on the one hand, while facing the adherent need of test routines, integration tests and penetration scans on the other hand. In fact, as the use of conventional methods are prohibitively expensive, better suited approaches are required which pinpoint attention to potentially problematic areas and obtain estimates of residual risk within a reasonable time frame. Solutions therefore cannot be only manually driven, but an orchestration of automated and manual tasks to discover test based deficiencies by a continuous approach.

In this section we give an overview of approaches to continuous security risk assessment that make use of different kinds techniques and tools, including security testing by monitoring and active testing. In Section 4.1 we describe approaches to continuous security risk management in the organizational context and at enterprise-level, and in Section 4.2 we present two relevant standards on the use of security indicators to support security monitoring. In Section 4.3 we give an overview of the state of the art on techniques and tools that give support for different kinds of continuous security risk assessment.

4.1 Continuous Risk Assessment at Enterprise-Level

Contrary to manually executed tasks, continuous event monitoring could facilitate continuous assessment by detecting certain events or patterns and feeding the results to the risk assessment. In this setting, automated processes can contribute to retaining an up-to-date view on the security risks by maintaining the validity of risk models and risk assessment results.

With respect to the security of ICT systems and software conformance, security information can either be derived from multiple test levels which cover the analysis of source code by highlighting defects and anomalies, or alternatively on the binary level were security scanners report discovered defects by number and severity. However, this highly sophisticated approach to continuous assessment requires automated script for the analysis and monitoring techniques to continuously detect event pattern and incidents.

Security risk monitoring, as supported by e.g. GRC frameworks such as the one depicted in Figure 1 in Section 2, may improve the transparency by the use of real-time response to risk levels or control exceptions. In industrial appliances, this is realized using a middleware layer which monitors incoming events, tracks events of interest, or detects patterns that indicate the existence of a condition which could indicate a problem. This can range from simple filtering up to a complex set of rules that correlate incoming events [73].

Essentially two variants of complex event processing (CEP) exist. The rule-based processing is simply monitoring the event stream for a certain event, which in turn triggers actions. The more advanced form of CEP is based on data sets, and utilizes a continuous query to analyze the data streams. Hence data from multiple sources can be combined to gain richer and more complete information.

From the actions triggered and the information provided by the CEP, continuous information can be computed to enable a rapid decision-making and collect raw data into a historical database for post-processing and compliance. This information is then fed into the connecting software in charge the enterprise-level governance, risk management and compliance (GRC) [74]. Hereby, a full integration of business operations and GRC management is achieved. Due to real-time responses on system events, a detective and preventive GRC management can be set up using automated mechanisms through continuous control monitoring (CCM), continuous risk monitoring (CRM) or continuous exception monitoring (CEM).

This offers 100% coverage of all business process instances which are observed by a CEP engine, instead of manually drawing samples afterwards. The resulting very short reaction time using real-time event detection has two strong advantages: First of all, as opposed to monitoring in discrete steps, continuous monitoring measures business reality instead of documented to-be processes. Secondly, events are captured at the time they occur, and not only before audits or other specified periodical times.

In fact, implementing continuous monitoring offers more than control automation and increased speed. Using the Classic GRC approach [75] as depicted to the left in Figure 3, business reality is manually documented in office documents which might take a lot of time. These artifacts, which might be outdated already by the times they are composed, always impede a time delay which will affect all subsequent steps. Following up from Internal Control Systems (ICS), reports are generated which are considered in the ICS Planning phase. As mentioned before, the underlying data is a snapshot based on the data documented before and may also be incomplete. With an additional delay, reports are composed for the GRC Management which needs to steer and react on its outcome.

The Real GRC approach using real-time continuous monitoring, as depicted to the right in Figure 3, is different in many ways. By using many in-place adapters, the business reality is automatically monitored. The use of automated controls, flexible query scripts and ICS Reporting can be conducted as automated "push-button tasks". Such automation can also monitor exceptions of critical process steps, as well as risk & fraud indicators. Results based on real data can in the ICS Planning phase be depicted at real-time providing a fully integration and fully automated test cases. This gives the GRC Management not only a better way to judge and take corresponding actions, but also mitigates possible sources of manual errors. Responses and corrective actions can occur in real time according to risk levels and control exceptions, and be preventive rather than detective. The combination of GRC and the complex event processing middleware layer allows full integration of business operations and GRC management.

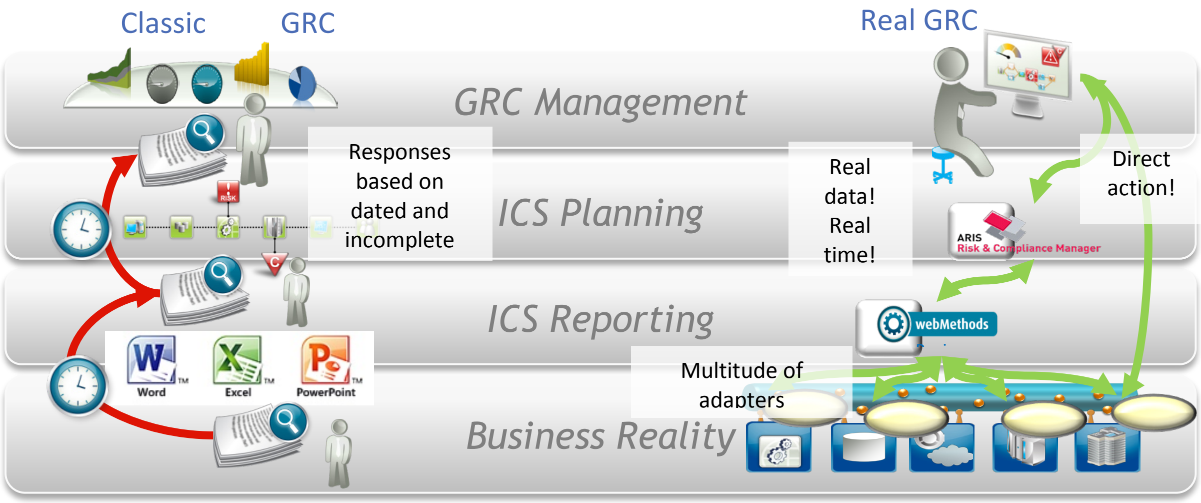

Figure 3: Real GRC and affected layers

4.2 Security Indicators for Organizations

For many organizations, it is crucial to prove or substantiate that they can offer and maintain a certain level of security and trustworthiness of the information systems and services they provide. Ideally, this should be supported by objective, measurable indicators and metrics that are evaluated regularly. Multiple sets of indicators and metrics have been proposed in the literature, and there are several relevant standards.

The purpose of the Information Security Management System (ISMS) as defined by ISO/IEC 27001 [21] is to provide organizations with best practices and guidelines for the entire information security management process, including the use of security indicators. The ISO/IEC 27004 [22] standard provides guidance on the use of measures and measurements for the assessment of the effectiveness of an ISMS. An abstract template for an information security measurement construct is provided along with some concrete examples of use. Such a construct consists of a general measurement description, the measure(s) specification and an indicator specification, which defines how the result will be represented. Furthermore, it defines what kind of action is required for any result, who is responsible, and who should see the results. The information security measurement constructs may support continuous analysis by explicitly defining how frequent data has to be collected and analyzed. Even the frequency for the measurement revision has to be specified.

The Information Security Indicators (ISI) specification [41] is currently being produced by an ETSI Industry Specification Group. The standard aims to assess the security posture of an organization. In contrast to ISMS, ISI is going to provide a catalogue of nearly 100 operational security indicators of both internal and external origin for measuring the effectiveness of an organization's security policy.

For each indicator, means and tools for the detection of relevant events are given, as well as the respective detectability level ranging from 1 (very difficult) to 3 (relatively easy). As additional information, the state-of-the-art detection ratio is expressed as the monthly frequency of the occurrence of an event or as the percentage against a common basis. Examples of indicator categories are indicators with security incidents (e.g. external intrusions and attacks), and indicators with vulnerabilities of a behavioral and technical kind (e.g. software vulnerabilities and configuration vulnerabilities).

4.3 Techniques for Continuous Risk Assessment

Various techniques can be used to support continuous risk assessment in order to ensure that the risk picture is kept valid and up to date while systems evolve. Some of these techniques have the potential to support continuous assessment at enterprise-level as discussed in Section 4.1, while others can make use of measures and indicators such as those presented in Section 4.2.

In addition to techniques for modular risk assessment as discussed in Section 2, approaches have been developed to facilitate traceability of changes from the target systems to the risk models, while others have been developed to enable continuous risk assessment by means of monitoring. In the following we give an overview of the latter two topics.

4.3.1 Traceability

Some of the established techniques for risk and threat modeling facilitate automatic updating of the values that are annotated on the diagrams; by changing input values to capture changes in the target of analysis, the derived output values can be generated. These techniques include fault trees, Markov models and Bayesian networks. Influence diagrams [58] were originally a graphical language designed to support decision making by specifying the factors influencing a decision. In [55], such diagrams are connected to the leaf nodes of fault trees supporting the propagation of influence to the unwanted incidents specified at the root of the tree. Similar, but simpler, are the risk influence diagrams, detailed in [48], where influencing factors are connected to the nodes in event trees. Several other notations have support for associating elements of risk models to parts of the target description, which may facilitate the identification of possible risk changes due to target changes. Approaches based on the UML, such as misuse cases [70], may utilize built-in mechanisms in the UML for relating elements from different UML diagrams that serve as the target model.

In [65] the risk graph notation is extended with support for the explicit modeling of risk changes. When systems change, some risks emerge and some become obsolete, while other risks are modified, and the extended risk graph notation has support for capturing all these aspects. The notation is moreover extended with support for relating risk graph elements to elements of the target model that is developed during the initial context establishment of a risk assessment process. The target model is a specification of the elements of the target of analysis, including software and hardware components, users and roles, information and communication networks, business and work processes, and so forth. The target model is created using a suitable notation such as activity diagrams, class diagrams, data flow diagrams or business process modeling. The specification of the relations between the target model and the risk model is in [65] referred to as the trace model, as it facilitates the systematic traceability of changes from the target model to the risk model.

By developing support for change management and traceability using risk graphs, the techniques can be instantiated by other existing techniques as explained in Section 2. In [65] the authors demonstrate this with an instantiation in CORAS, exemplifying the technique by applying CORAS threat diagrams to a case study from the air traffic management domain. The tool presented in [69] is developed to support this instantiation in CORAS. The main feature of the tool is the diagram editor for creating all kinds of CORAS diagrams to model and assess changing and evolving risks. The tool moreover supports the specification of the trace model and the automation of traceability. The approach does not require a specific notation to be used for the target modeling. Instead, the target model is specified in a language of choice and in a separate tool of choice, for example using UML, business process modeling, requirement models, or any notation that is suitable and adequate for the target of analysis and the desired level of abstraction. Still, because the CORAS tool is developed as a plugin to Eclipse, the meta-model of the target model must be an (Eclipse) Ecore meta-model.

Thales Research & Technology has developed their own industrial model-based approach to risk assessment, supported by the Rinforzando [68] tool. The security risk assessment and modeling can be performed as stand-alone, but is also designed to serve as an integrated part of their mainstream system engineering workbench [72]. For this purpose, dynamic links can be built and maintained between the risk models and the system engineering models, the latter specified using a service oriented architecture (SOA) modeling suit. When any model changes are implemented during the system development process, either on the risk model or the system model, the changes are immediately propagated via the links to trigger updates and maintain the mutual consistency between the modeling domains. The integration with their system engineering process is hard-coded and much tighter than what is offered by CORAS. However, this is at the cost of general applicability as Rinforzando is tied to the Thales engineering workbench, whereas CORAS allows any notation to be used for target modeling.

Model Versioning and Evolution (MoVE) [66][50] is an approach to build an infrastructure to maintain the validity, mutual consistency and interdependencies between models as they evolve over time within model-driven engineering (MDE). The approach does not target security and risk in particular, but rather builds a tool-supported infrastructure for versioning of several interdependent models, for example for software architecture and design, business processes, services, security and risk. Similar to the aforementioned tools, the underlying idea is to provide support for tracing changes from one model to another to ensure that they are globally up-to-date and mutually consistent. However, although the infrastructure can support the handling of traceability and evolution in security risk analysis, as exemplified in a case study, there is no specific modeling or methodological support for this. Instead the user chooses the models and notations, such as event trees, CORAS, data flow diagrams, UML, etc., to be managed by MoVE.

The model-based approach to risk analysis with support for dependency identification and modeling presented in [59] relates risk elements to elements of a functional model of the target of analysis. Moreover, the model elements are related to security objectives and security requirements, and risks are related to threats and security controls. Although risk assessment is supported, there is no risk modeling support other than a simple, high-level description of incidents and their likelihood and consequence.

4.3.2 Monitoring

Risk monitoring is a means to facilitate continuous risk assessment by the monitoring of relevant key indicators or metrics. An indicator can be defined as "something that provides a clue to a matter of larger significance or makes perceptible a trend or phenomenon that is not immediately detectable" [57]. For example, an unexpected rise in traffic load of a web server may signal a denial of service attack in progress. In order to enable security risk monitoring based on the monitoring of key indicators or metrics, there is a need not only to identify the relevant indicators, but also to understand how to relate the indicators to potential security risks, and how to aggregate the monitored values into risk levels. The benefit of security risk monitoring is, of course, that risk assessment results can be automatically updated as they evolve while the target of analysis evolves.

The work presented in [67] is a model-based approach to make use of measurable indicators in order to obtain a risk picture that is continuously or periodically updated. The approach comes with a process of three steps. First, an initial risk analysis is conducted to identify and model possible threat scenarios and unwanted incidents. Second, key indicators are identified that may be relevant for determining likelihoods and consequences for the risk model. Third, functions are defined for calculating likelihoods and consequences based on the indicators. Using this model-based approach, managers and other stakeholders are provided a high-level view of the current system security by observing the updates of the risk models.

Closely related to this notion of indicators is the notion of ICT security metrics for measuring information security. The NIST Performance Measurement Guide for Information Security [51] aims to assist in the development, selection, and implementation of suitable measures to this end. The guide moreover provides several candidate measures, such as the percentage of information system security personnel that have received security training, or the percentage of individuals screened before being granted access to organizational information and information systems. Unlike the approach in [67], the NIST guidelines do not necessarily aim to establish explicit frequency, consequence, and risk levels from the identified set of metrics.

The approach proposed in [62] focuses on the security of dynamic services in the more complex setting of systems of systems. The latter are collections of systems interconnected through the exchange of services. The authors propose a method to support the capturing and the monitoring of the impact of service dependencies on the security of the provided services. The method is divided into four main steps: i) documenting the system of systems and ICT service dependencies, ii) establishing the impact of service dependencies on the security risk of provided services, iii) identifying measureable indicators for dynamic monitoring, and iv) specifying the indicator design and use. In a different publication [63], the same authors address the related problem of designing the indicators to be monitored. For the security risk monitoring to be correct, it is of course crucial that the selected indicators provide a valid view of the risk picture and the monitored risk level. A method is presented and exemplified for identifying relevant key indicators and evaluating their validity.

Adequate tool-support is obviously a necessity for enabling security risk monitoring and the continuous aggregation of measured indicator values to generate the updated risk levels. Within business intelligence (BI) there exist several tools that support enterprise level monitoring. Digital dashboards [54] visualize monitored data to provide an intuitive representation, allowing managers to monitor the progress towards identified enterprise level goals. Data mining [56] uses techniques from statistics and artificial intelligence to identify patterns in large set of business data. Process mining [47] extracts information about business process using event logs. Both data and process mining use historical data as input. In [64], an architectural pattern for enterprise level monitoring tools that uses both real-time and historical data is proposed. The idea is that the pattern should serve as a generic basis for building tools with features for collecting low-level data from the ICT infrastructure, aggregating the collected low-level data, evaluating the aggregated data, presenting the aggregated data and the evaluation results to different stakeholders, as well as features for doing the necessary configurations. The architectural pattern is not limited to tools for security risk monitoring, but the authors demonstrate its use by the implementation of a run-time risk monitor using CORAS for risk modeling. MASTER ESB [52] is a monitoring tool closely related to this pattern. The tool is for the monitoring of compliance with access and usage policies by monitoring low-level data that is aggregated and evaluated against the specified policies.

Krautsevich et al. [60][61] propose an approach to make use of run-time attribute monitoring to support risk-based enforcement of usage control (UCON) policies. Some existing approaches [49][53][71] already propose the use of risk assessment to support access control, but focus only on the problem of authorization prior to granting access. Krautsevich et al., on the other hand, stress the dynamic nature of UCON where authorization may change over time. Because UCON decisions are based on mutable attributes, the values of which evolve, the reference monitor continuously needs to reevaluate the enforcement decisions by continuously reassessing the involved risks.

In the envisaged RASEN framework, we will leverage traceability techniques to support continuous security risk assessment, since this is an efficient means of linking parts and aspects of the target of analysis to the risk models that need to be kept valid and up to date. Moreover, with our approach to test-based risk assessment, support for traceability will in particular be developed for linking test results to the continuous risk assessment activities. The testing will include not only monitoring, as discussed in this section, but also active testing as discussed in Section 0. Additionally, and more important, RASEN will strongly leverage our techniques for compositional security risk assessment to support continuous risk assessment. Because compositional assessment is not well-supported by the current state of the art, existing techniques for continuous risk assessment generally make use of variants of traceability and monitoring. Adequate techniques and tools for compositionality, on the other hand, have the potential to strongly facilitate continuous assessment by the support for rapidly reassessing only the parts of the system that have changed.

5 RASEN Baseline

In this section we recapitulate by describing the baseline for the upcoming RASEN R&D activities of WP3. The baseline consists of state of the art techniques and tools that we believe are a good starting point for addressing the research questions that are relevant for this work package. The section is structured according to the state of the art sections of this deliverable, which also corresponds to the three main research tasks of WP3.

5.1 Baseline for Compositional and Modular Security Risk Assessment

One of the research questions that RASEN addresses is how to use composition to make security assessment of large scale networked systems more feasible. As discussed in Section 2, there is very little support for compositional risk modeling and assessment in existing approaches and techniques. We envisage that such techniques would substantially facilitate risk assessment of large systems by decomposing the target system into smaller, manageable parts. As part of the RASEN baseline we will build on established and well known techniques for risk modeling. We aim to develop novel principles and techniques that as much as possible can be instantiated by existing notations such as ETA, FTA, Bayesian networks, cause-consequence diagrams, etc. For this purpose we will use risk graphs [5] as the starting point and baseline for compositional risk modeling. Risk graphs are defined by a formal syntax and semantics, and also come with a calculus for precise and rigorous analysis. Moreover, risk graphs can be understood as a common abstraction of several established risk modeling techniques, which means that the risk graph formalization and its calculus can be instantiated by these. In RASEN we will extend the formalization of risk graphs to enable modularity. We also aim to demonstrate the application of these techniques in possible instantiations, such as CORAS. These instantiations will moreover be evaluated and validated in the industrial case studies.

When developing techniques for modularity and compositional assessment, we will carefully consider existing work such as dependent risk graphs and Dependent CORAS [28], as well as the CORAS extension in [46] to support reuse of risk analysis results. In this setting, the RASEN project will also investigate another relevant research question, namely to what extent criteria for valid composition of risk assessment results can be related to valid criteria for composition of security test results. Such criteria are needed to make efficient use of test-based risk assessment in a compositional setting. As part of the RASEN baseline, we will explore the potential of leveraging traceability techniques similar to those presented in Section 0. By building such traceability to relate test results and risk models, there may be a potential to extend the approach with criteria for valid composition of both risk results and test results, as well as the traceability between them.

5.2 Baseline for Test-Based Risk Identification and Estimation

As discussed in Section 0, we are only aware of three approaches that provide some support for test-based security risk assessment. The approach of Wong et al. [84] is only applicable to source code analysis, which is too restrictive considering the challenges addressed by RASEN. The approach of Erdogan et al. [82] is based on precise risk models, but no technique for updating the risk model based on test results is proposed. The approach of Benet el. al. [81] does describe how the test results can impact the risk picture, but their notion of risk model is imprecise.

The approach of Erdogan et al. and Benet complement each other in the sense that the strength of one approach is a weakness of the other and vice versa. A part of the RASEN baseline with respect to test-based security risk assessment will therefore be the combination of these two approaches. This will involve applying the ideas of Benet on how test results impact the risks in the setting of CORAS risk models [28]. To achieve this, the CORAS risk model language may have to be extended with new language constructs. Such an extension will also be developed for risk graphs to ensure applicability of our techniques beyond CORAS.

In the context of complex networked systems, application security risks are nowadays among the most prominent risks. As mentioned by in [85] "*Attacks have changed from noisy, mass attacks aimed at large numbers of systems to targeted and financially motivated attacks that are highly focused on manipulating applications and stealing or tampering with sensitive data. Hackers often use low-cost or* *free tools to scan for and find exploitable application vulnerabilities.* […] *The most critical impact of using application security is minimizing the risk of the possible exploitation of application vulnerabilities. Adopting application security will enable organizations to detect the vulnerabilities embedded in applications before hackers detect them. Also, applying application security enables early detection of security vulnerabilities with early (and thus, less expensive) remediation.*"

In the RASEN test-based risk assessment approach, security testing will be used to detect potential vulnerabilities, using various techniques such as Dynamic Application Security Testing (Web vulnerability Scanners, Model-based vulnerability testing, Fuzzing) and Static Application Security Testing (code scanning, static analysis). These automated testing techniques provide as results vulnerability detection, including impact metrics and reference to vulnerability databases (such as CVE [86] or OWASP [89]). Moreover, securing testing will help to discover new potential vulnerabilities that have been identified in the previous risk identification phase.

Figure 4 presents a first picture of the RASEN approach to test-based risk identification and estimation. In this approach, security testing interacts with all phases of the risk assessment (risk identification, risk analysis and risk estimation), but also with the risk treatment phase. Here, by security testing, we mean mainly automated vulnerability testing using techniques and tools such as web vulnerability scanners, model-based vulnerability testing tools and fuzzing techniques. These techniques are presented in more details in RASEN Deliverable D4.1.1.

The possible influence of security testing activities on risk assessment includes the following:

- **Risk identification**. It is almost impossible to have a comprehensive identification of existing potential vulnerabilities in a web application without testing it. Vulnerability databases, such as CVE [86], and forums help, but this is obviously related to already disclosed vulnerabilities. Application security testing may discover further vulnerabilities, thereby contributing to the security risk identification.

- **Risk analysis**. One common problem of automated application security testing tools is their potential to provide false positives (see for example [90] for web application vulnerability scanners). Therefore, risk analysis will investigate the potential impact of each of the detected vulnerabilities by security testing.

- **Risk evaluation**. In order to determine the significance of the detected vulnerabilities after the security testing and the risk analysis phase, a ranking can be specified by using an estimation approach such that the Common Vulnerability Scoring System (CVSS) [88]. The CVSS provides an open framework for communicating the characteristics and impacts of ICT vulnerabilities.

- **Risk treatment**. Depending of risk assessment phases, remediation actions may be decided and implemented. Then, security testing is used to confirm the mitigation of identified risks.

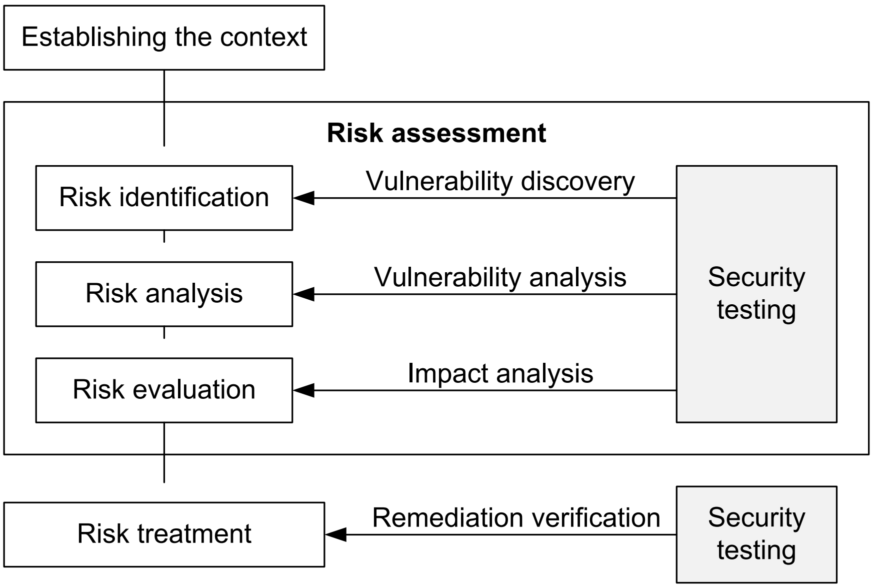

Figure 4 - RASEN approach for test-based risk assessment and treatment

5.3 Baseline for Continuous Security Risk Assessment

Today, many enterprises rely on some kind of continuous assessment of information, monitoring of pattern and the identification of risk. Over the past years supporting software and tool chains evolved and stabilized, providing a sound basis upon which risk assessment can be conducted. However, as networked ICT systems and services become larger and more complex, heterogeneous and dynamic, the problem of continuously maintaining a valid risk picture becomes increasingly challenging.

In addressing these challenges, we will build on existing techniques for traceability (e.g. [50][65][69]) to facilitate rapid identification of risks that need to be reassessed given changes in the target of analysis. Traceability is moreover an adequate means for maintaining the mutual consistency between model artifacts, for example between risk models and target models, or risk models and test models. We will also build on existing techniques for test-based risk identification, which means that the RASEN baseline in this respect, as described above, is relevant also for continuous security risk assessment.

Furthermore, in the context of the RASEN project, we will utilize our own techniques for compositional risk assessment to facilitate continuous risk assessment. Hence, in the course of the project, the WP3 task of developing tools and techniques for compositional assessment largely precedes the task of developing support for continuous assessment. As part of the common RASEN WP3 baseline, we will use as starting point the risk graph notation to do continuous risk modeling and to build the traceability links between the risk models and related model artifacts.

6 Summary

In the information society of today, networked ICT systems and services become ever more complex, heterogeneous and dynamic. Moreover, networked service and computing environments cross organizational and geographic borders, adding further to the complexity of managing security and assessing the involved security risks.

RASEN WP3 aims to provide means to tackle these challenges by developing novel tools and techniques for compositional security risk assessment. Compositional techniques should allow complex systems to be assessed by addressing small and manageable parts of the system in isolation, for later composing the risk assessment results of each part to generate the global risk picture. Compositionality should also facilitate continuous risk assessment by the support for reassessment of only the parts of the system that have changed.

A further challenge addressed by RASEN is how to systematically combine high-level security risk assessment with low-level security testing. The objective of WP3 regarding this challenge is to develop techniques and tools for test-based security risk assessment.

In this deliverable we have given an overview of relevant state of the art, structured according to the three main research tasks of WP3. Task T3.1 is on compositional security risk assessment, the state of the art for which is described in Section 2. Task T3.2 is on test-based risk identification and estimation, which is described in Section 0. Task T3.3 is on continuous risk assessment of large scale system by use of test-based indicators, which is described in Section 0.

As a summary, we briefly conclude in the following on each of the relevant RASEN research questions listed in the introduction:

- *To what extent can we use composition to make security assessment of large scale networked systems more feasible and understandable?* As discussed in Section 2, techniques for modularity and decomposition are virtually non-existent within risk management and security risk assessment. There exists some support for component-based risk assessment, where individual components can be assessed in isolation, but there is still a need for techniques and tools to support systematic composition and decomposition of risk models for large networked ICT systems and services in general.

- *To what extent can we relate criteria for valid composition of security risk assessment results to criteria for valid composition of security test results?* This research question is crucial given the RASEN objective of developing support for combining test-based security risk assessment and compositional risk assessment. As discussed in Section 2 and Section 0, this is a completely open research question given the state of the art. First, adequate and efficient criteria for compositional risk assessment alone pose a timely research question. Second, relating such criteria to criteria for composition of security test results require better techniques for enabling test-based risk assessment than what is offered by the current state of the art.

- *To what extent can we support reuse of security risk assessments to make security assessment of large scale networked systems more feasible?* As discussed in Section 0, there exist some support for continuous risk assessment that facilitate reuse in the sense that risk assessment results that are not affected by change should not be analyzed anew. In the context of the RASEN project, novel techniques for risk composition will serve as the most important means for reuse of assessment results.

- *What are good methods and tools for aggregating test results—obtained by both active testing and passive testing (monitoring)—to the risk assessment?* Test-based risk assessment requires means for how to make use of test results at the risk assessment level. For this research question, there many existing techniques, tools, guidelines and standards that can be utilized, as discussed in Section 0 and Section 0. In WP3 we will leverage such techniques while adapting them to support the overall RASEN tool-supported framework for test-based security risk assessment.

- *How can test results be exploited to obtain a more correct risk picture?* This research question is more general than the above ones in the sense that there are a wide range of different approaches to make use of testing to support risk assessment. In RASEN WP3 we will build on existing techniques, but most importantly address this research question by investigating the support that we develop within the RASEN framework as a whole. In particular, a guiding principle for the WP3 R&D work is that our techniques and tools for compositional and test-based security risk assessment indeed should contribute to obtaining and maintaining a correct risk picture for large and complex networked systems and services.

References

[1] C. J. Alberts, A. J. Dorofee: OCTAVE Criteria. Technical Report CMU/SEI-2001-TR-016, CERT (2001)

[2] Ben-Gal: Bayesian networks. In: F. Ruggeri, R. S. Kenett, F. W. Faltin (eds.): Encyclopedia of Statistics in Quality and Reliability. John Wiley & Sons (2007)

[3] Bouti, A. D. Kadi: A state-of-the-art review of FMEA/FMECA. International Journal of Reliability, Quality and Safety Engineering 1, 515–543 (1994)

[4] M. Broy and K. Stølen: Specification and Development of Interactive Systems: Focus on Streams, Interfaces and Refinement. Springer (2001)

[5] G. Brændeland, A. Refsdal, K. Stølen: Modular analysis and modelling of risk scenarios with dependencies. Journal of Systems and Software 83(10), 1995–2013 (2010)

[6] G. Brændeland, K. Stølen: Using model-driven risk analysis in component-based development. In: Dependability and Computer Engineering: Concepts for Software-Intensive Systems, pp. 330-380, IGI Global (2011)

[7] V. Cortellessa, K. Goseva-Popstojanova, K. Appukkutty, A. Guedem, A. E. Hassan, R. Elnaggar, W. Abdelmoez, H. H. Ammar: Model-based performance risk analysis. IEEE Transactions on Software Engineering, 31(1), 3–20 (2005)

[8] N. Fenton and M. Neil: Combining evidence in risk analysis using Bayesian networks. Agena White Paper W0704/01, Agena (2004)

[9] H. Giese and M. Tichy: Component-based hazard analysis: Optimal designs, product lines, and online-reconfiguration. In: Computer Safety, Reliability, and Security (SAFECOMP'06). LNCS, vol. 4166, pp. 156–169. Springer (2006)

[10] H. Giese, M. Tichy, and D. Schilling. Compositional hazard analysis of UML component and deployment models. In: Computer Safety, Reliability, and Security (SAFECOMP'04). LNCS, vol. 3219, pp. 166–179. Springer (2004)

[11] K. Goseva-Popstojanova, A. E. Hassan, A. Guedem, W. Abdelmoez, D. E. M. Nassar, H. H. Ammar, A. Mili: Architectural-level risk analysis using UML. IEEE Transactions on Software Engineering, 29(10), 946–960 (2003)

[12] S. Hernan, S. Lambert, T. Ostwald, A. Shostack: Threat modeling – uncover security design flaws using the STRIDE approach (2006) [ONLINE] Available at: http://msdn.microsoft.com/en-us/magazine/cc163519.aspx [Accessed December 3, 2012]

[13] R. A. Howard: Dynamic Probabilistic Systems, Volume I: Markov Models. John Wiley & Sons (1971)

[14] Humprey: SWOT: Strengths, Weaknesses, Opportunities, Threats. Stanford University (1960–1970)

[15] International Electrotechnical Commission: IEC 60300-3-9 Dependability management – Part 3: Application guide – Section 9: Risk analysis of technological systems – Event Tree Analysis (ETA) (1995)

[16] International Electrotechnical Commission: IEC 61025 Fault Tree Analysis (FTA) (1990)

[17] International Electrotechnical Commission: IEC 61165 Application of Markov Techniques (1995)

[18] International Electrotechnical Commission: IEC 61882 Hazard and Operability studies (HAZOP studies) – Application guide (2001)

[19] International Organization for Standardization: ISO 31000 Risk management – Principles and guidelines (2009)

[20] International Organization for Standardization: ISO Guide 73 Risk management – Vocabulary (2009)

[21] International Organization for Standardization / International Electrotechnical Commission: ISO/IEC 27001 Information technology – Security techniques – Information security management systems – Requirements (2005)

[22] International Organization for Standardization / International Electrotechnical Commission: ISO/IEC 27004 Information technology – Security techniques – Information security management – Measurement (2009)

[23] International Organization for Standardization / International Electrotechnical Commission: ISO/IEC 27005 Information technology – Security techniques – Information security risk management (2011)

[24] International Organization for Standardization / International Electrotechnical Commission: ISO/IEC 31010 Risk management - Risk assessment techniques (2009)

[25] B. Kaiser, P. Liggesmeyer, and O. Mäckel: A new component concept for fault trees. In: 8th Australian workshop on Safety critical systems and software (SCS'03), pp. 37–46. Australian Computer Society (2003)

[26] H. A. Linstone, M. Turoff: The Delphi Method - Techniques and Applications. Addison-Wesley (1975)

[27] M. S. Lund, B. Solhaug, K. Stølen: Evolution in relation to risk and trust management. Computer 43(5), 49–55 (2010)

[28] M. S. Lund, B. Solhaug, K. Stølen: Model-Driven Risk Analysis – The CORAS Approach. Springer (2011)

[29] S. Mannan (ed.): Lees' Loss Prevention the Process Industries. Hazard Identification, Assessment and Control, vol. 1, 3rd edn. Elsevier/Butterworth/Heinemann (2005)

[30] Object Management Group: OMG Unified Modeling Language (OMG UML), Superstructure. Version 2.4. OMG Document Number: ptc/2010-11-14 (2011)

[31] Y. Papadoupoulos, J. McDermid, R. Sasse, and G. Heiner: Analysis and synthesis of the behaviour of complex programmable electronic systems in conditions of failure. Reliability Engineering and System Safety, 71(3), 229–247 (2001)

[32] R. M. Robinson, K. Anderson, B. Browning, G. Francis, M. Kanga, T. Millen, C. Tillman: Risk and Reliability. An introductory text. 5th edition. Risk & Reliability Associates (2001)

[33] B. Schneier: Attack trees: Modeling security threats. Dr. Dobb's J. 24(12), 21–29 (1999)

[34] S. Sutton: Process Risk and Reliability Management: Operational Integrity Management. Elsevier (2010)

[35] F. Swiderski, W. Snyder: Threat Modeling. Microsoft Press (2004)

[36] D. Verdon and G. McGraw: Risk analysis in software design. IEEE Security & Privacy, 2(4), 79–84 (2004)

[37] J. M. Wing. A specifier's introduction to formal methods. IEEE Computer 23(9), 8,10–22,24 (1990)

[38] A. Kline, O. Renn: A new approach to risk evaluation and management: Risk-based, precaution-based and discourse-based strategies. Risk Analysis 22(6), 1071-1094 (2002)

[39] Z. Bluvband, R. Polak, P. Grabov: Bouncing failure analysis (BFA): The unified FTA-FMEA methodology. In: Annual Reliability and Maintainability Symposium (RAMS'05), pp. 463-467. IEEE (2005)

[40] J. Bechta Dugan, K. J. Sullivan, D. Coppit: Developing a low-cost high-quality software tool for dynamic fault-tree analysis. IEEE Transactions on Reliability 49(1), 49-59 (2000)

[41] ETSI: Information Security Indicators (ISI), Part 1 – A full set of operational indicators for organizations to use to benchmark their security posture, Version 0.0.2. ETSI (2012)

[42] R. Gulati, J. B. Dugan: A modular approach for analyzing static and dynamic fault trees. In: Annual Reliability and Maintainability Symposium (RAMS'97), pp. 57-63. IEEE (1997)

[43] A. Rauzy: New algorithms for fault trees analysis. Reliability Engineering & System Safety 40(3), 203-211 (1993)

[44] M. Stamatelatos, W. Vesley, J. Dugan, J. Fragola, J. Minarick, J. Railsback: Fault tree handbook with aerospace applications. Version 1.1. NASA Office of Safety and Mission Assurance (2002)

[45] W. E. Vesely, F. F. Goldberg, N. H. Roberts, D. F. Haasl: Fault Tree Handbook. U.S. Nuclear Regulatory Commission (1981)

[46] J. Viehmann: Reusing risk analysis results - An extension for the CORAS risk analysis method. In: 4th International Conference on Information Privacy, Security, Risk and Trust (PASSAT’12), pp. 742-751. IEEE (2012)

[47] W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, A. J. M. M. Weijters: Workflow mining: A survey of issues and approaches. Data Knowledge Engineering 47(2), 237–267 (2003)

[48] T. Aven, S. Sklet, J. E. Vinnem: Barrier and operational risk analysis of hydrocarbon releases (BORARelease). Part I. Method description. J. Haz. Mat. A137, 681–691 (2006)

[49] B. Aziz, A. N. Foley, J. Herbert, G. Swart: Reconfiguring role based access control policies using risk semantics. J. High Speed Networks 15(3), 261–273 (2006)

[50] M. Breu, R. Breu, S. Löw: MoVEing forward: Towards an architecture and processes for a Living Models infrastructure. International Journal on Advances in Life Sciences 3(1-2), 12-22 (2011)

[51] E. Chew, M. Swanson, K. Stine, N. Bartol, A. Brown, W Robinson: Performance measurement guide for information security. NIST Special Publication 800-55, revision 1. Technical report, National Institute of Standards and Technology (2008)

[52] B. Crispo, G. Gheorghe, V. D. Giacomo, and D. Presenza: MASTER as a Security Management Tool for Policy Compliance. In: Towards a Service-Based Internet. LNCS, vol. 6481, pp. 213–214. Springer (2010)

[53] N. N. Diep, L. X. Hung, Y. Zhung, S. Lee, Y. K. Lee, H. Lee: Enforcing access control using risk assessment. In: 4th European Conference on Universal Multiservice Networks (ECUMN’07), pp. 419–424. IEEE Computer Society (2007)

[54] C. Dover: How dashboards can change your culture. Strategic Finance 86(4), 43–48 (2004)

[55] EUROCONTROL: Methodology report for the 2005/2012 integrated risk picture for Air Traffic Management in Europe. EEC Technical/Scientific Report No. 2006-041 (2006)

[56] Y. Fu: Data Mining. IEEE Potentials 16(4), 18–20 (1997)

[57] A. Hammond, A. Adriaanse, E. Rodenburg, D. Bryant, R. Woodward: Environmental Indicators: A Systematic Approach to Measuring and Reporting on Environmental Policy Performance in the Context of Sustainable Development. World Resources Institute (1995)

[58] R. A. Howard, J. E. Matheson: Influence diagrams. Decis. Anal. 2(3), 127–143 (2005)

[59] F. Innerhofer-Oberperfler, R. Breu: Using an enterprise architecture for IT risk management. In: Information Security South Africa Conference (ISSA’06), (2006)

[60] L. Krautsevich, A. Lazouski, F. Martinelli, A. Yautsiukhin: Risk-aware usage decision making in highly dynamic systems. In: 5th International Conference on Internet Monitoring and Protection (ICIMP'10), pp. 29-34. IEEE Computer Society (2010)

[61] L. Krautsevich, A. Lazouski, F. Martinelli, A. Yautsiukhin: Cost-effective enforcement of UCONA policies. In: 6th International Conference on Risks and Security of Internet and Systems (CRiSIS'11), pp. 1-8. IEEE Computer Press (2011)

[62] O. S. Ligaarden, A. Refsdal, K. Stølen: Using indicators to monitor security risk in systems of systems: How to capture and measure the impact of service dependencies on the security of provided services. In: IT Security Governance Innovations: Theory and Research, pp. 256-2922. IGI Global (2012)

[63] O. S. Ligaarden, A. Refsdal, K. Stølen: Designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators. International Journal on Advances in Intelligent Systems 5(1-2), 173-195 (2012)

[64] O. S. Ligaarden, M. S. Lund, A. Refsdal, F. Seehusen, K. Stølen: An architectural pattern for enterprise level monitoring tools. In: Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA'11), pp. 1-10. IEEE Computer Society (2011)

[65] M. S. Lund, B. Solhaug, K. Stølen: Risk analysis of changing and evolving systems using CORAS. In: Foundations of Security Analysis and Design VI (FOSAD VI). LNCS, vol. 6858, pp. 231–274. Springer (2011)

[66] MoVE: Model versioning and evolution. [ONLINE] Available at http://move.q-e.at/ [Accessed 4 December 2012]

[67] A. Refsdal, K. Stølen: Employing key indicators to provide a dynamic risk picture with a notion of confidence. In: Trust Management III. IFIP Advances in Information and Communication Technology, vol. 300, pp. 215–233. Springer (2009)

[68] SecureChange: Integrability of design modelling solution. SecureChange project deliverable D4.4b (2011)

[69] F. Seehusen, B. Solhaug: Tool-supported risk modeling and analysis of evolving critical infrastructures. In: CD-ARES 2012. LNCS, vol. 7465, pp. 562–577. Springer (2012)

[70] G. Sindre, A. L. Opdahl: Eliciting security requirements by misuse cases. In: 37th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS Pacific'00), pp. 120–131. IEEE Computer Society (2000)

[71] C. Skalka, X. S. Wang, P. Chapin: Risk management for distributed authorization. Journal of Computer Security 15(4), 447-489 (2007)

[72] J. L. Voirin: Method & tools for constrained system architecting. In: 18th Annual International Symposium of the International Council on Systems Engineering (INCOSE'08), pp. 775-789. Curran Associates, Inc. (2008)

[73] T. Hugh: CEP for Dummies®, SAP Special Edition. John Wiley & Sons (2012)

[74] ARCM, Software AG, ARIS Risk & Compliance Manager, ARIS Risk & Compliance Manager Fact Sheet, Software AG (2011)

[75] ARCM, Business Solution, Governance, Risk & Compliance, Software AG, Product Marketing (2012)

[76] R. Angeli, M. Kling: Process risk simulation – Analyzing the effects of risks and controls in business processes. Business white paper, Software AG (2011)

[77] Software AG: Bow Tie method – User concept. Bow Tie methodology with Software AG, Governance, Risk and Compliance solution (2012)

[78] R. Davis: ARIS Design Platform – Advanced Process Modelling and Administration. Springer (2008)

[79] M. Hoffmann: Governance, risk and compliance (GRC) – An integrated approach. Business white paper, Software AG (2010)

[80] A.-W. Scheer: ARIS — Business Process Frameworks, 3rd ed. Springer (1999)

[81] A. F. Benet: A risk driven approach to testing medical device software. In: Advances in Systems Safety, pp. 157–168. Springer (2011)

[82] G. Erdogan, F. Seehusen, K. Stølen, J. Aagedal. Assessing the usefulness of testing for validating the correctness of security risk models based on an industrial case study. To

appear in Proc. International Workshop on Quantitative Aspects in Security Assurance (QASA'12).

[83] N. Kumar, D. Sosale, S. N. Konuganti, and A. Rathi: Enabling the adoption of aspects – testing aspects: A risk model, fault model and patterns. In: 8th ACM international conference on Aspect-oriented software development (AOSD'09), pp. 197–206. ACM (2009)

[84] W.E. Wong, Y. Qi, and K. Cooper: Source code-based software risk assessing. In: 2005 ACM symposium on Applied computing (SAC'05), pp. 1485–1490. ACM (2005)

[85] J. Feiman: Hype cycle for application security, 2012. GARTNER (2012)

[86] CVE – Common Vulnerabilities and Exposures. [ONLINE] Available at http://cve.mitre.org/ [Accessed 22 January 2013]

[87] Magento [ONLINE] Available at http://www.magentocommerce.com/ [Accessed 22 January 2013]

[88] National Vulnerability Database – NVD Common Vulnerability Scoring System Support v2. [ONLINE] Available at http://nvd.nist.gov/cvss.cfm?version=2 [Accessed 22 January 2013]

[89] The Open Web Application Security Project (OWASP) [ONLINE] Available at www.owasp.org/ [Accessed 22 January 2013]

[90] A. Doupé, M. Cova, and G. Vigna: Why Johnny can't pentest: An analysis of black-box web vulnerability scanners. In: Int. Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'10). LNCS, vol. 2460, pp. 111-131. Springer (2010)