EURO Journal on Decision Processes manuscript No.
(will be inserted by the editor)

Selecting security control portfolios: A multi-objective simulation-optimization approach

Elmar Kiesling · Andreas Ekelhart · Bernhard Grill · Christine Strauss · Christian Stummer

Abstract Organizations' information infrastructures are exposed to a large variety of threats. The most complex of these threats unfold in stages, as actors exploit multiple attack vectors in a sequence of calculated steps. Deciding how to respond to such serious threats poses a challenge that is of substantial practical relevance to IT security managers. These critical decisions require an understanding of the threat actors – including their various motivations, resources, capabilities, and points of access – as well as detailed knowledge about the complex interplay of attack vectors at their disposal. In practice, however, security decisions are often made in response to acute short-term requirements, which results in inefficient resource allocations and ineffective overall threat mitigation. The decision support methodology introduced in this paper addresses this issue. By anchoring IT security managers' decisions in an operational model of the organization's information infrastructure, we provide the means to develop a better understanding of security problems, improve situational awareness, and bridge the gap between strategic security investment and operational implementation decisions. To this end, we

Elmar Kiesling
Vienna University of Technology, Institute of Software Technology and Interactive Systems,
Favoritenstraße 9-11, Vienna, Austria
E-mail: elmar.kiesling@tuwien.ac.at

Andreas Ekelhart
Secure Business Austria, Favoritenstraße 16, Vienna, Austria
E-mail: aekelhart@sba-research.org

Bernhard Grill
Secure Business Austria, Favoritenstraße 16, Vienna, Austria
E-mail: bgrill@sba-research.org

Christine Strauss
University of Vienna, Faculty of Business, Economics, and Statistics,
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
E-mail: christine.strauss@univie.ac.at

Christian Stummer (Corresponding author)
Bielefeld University, Department of Business Administration and Economics,
Universitätsstraße 25, 33615 Bielefeld, Germany
E-mail: christian.stummer@uni-bielefeld.de
Tel. +49 521.106-4892, Fax -154891

combine conceptual modeling of security knowledge with a simulation-based optimization that hardens a modeled infrastructure against simulated attacks, and provide a decision-support component for selecting from efficient combinations of security controls. We describe the prototypical implementation of this approach, demonstrate how it can be applied, and discuss the results of an in-depth expert evaluation.

Keywords IT security analysis · multi-objective portfolio selection · interactive decision support · simulation · genetic algorithm

Mathematics Subject Classification (2000) 68U20 · 68U35 · 90B50 · 90C27 · 91B32

1 Introduction

Most organizations today rely heavily on information systems in their daily operations. Since the vast majority of companies have already encountered external as well as internal security incidents (Kaspersky, 2014), annual information security spending has risen sharply to more than USD 60 billion (Economist, 2014). Still, the annual costs to the global economy from cybercrime range between USD 375 and 575 billion (McAfee, 2014). Ensuring the security of the business-critical systems has therefore become a key management concern. However, assessing the effectiveness of these vast investments is challenging because of the difficulty to measure averted losses, the technical complexity involved, and the nature of security as a "moving target".

Information technology (IT) specialists responsible for ensuring the security of information systems usually focus on technical aspects. Based on implicit assumptions about hypothesized threats, they aim to identify and implement an appropriate set of measures that includes physical, technical, operational, and organizational security controls. However, security experts frequently struggle with justifying such investments in terms that senior managers can relate to. Resource allocation decisions, therefore, tend to be driven by immediate needs or a diffuse fear of potential losses. This reactive ad-hoc approach ultimately leads to inefficient resource allocations and unsatisfactory overall security.

Determining the "best" portfolio of security controls not only is of obvious practical relevance, but it also is of interest from a decision-making (research) point of view. This decision involves multiple stakeholders with diverse perspectives, requires trade-offs between conflicting objectives, and is characterized by substantial complexity and uncertainty. Furthermore, information infrastructures, the threats they are exposed to, and the decision makers' risk preferences differ greatly among organizations. The same holds for the attackers' motives, goals, skills, and points of access all of which determine the attack vectors at their disposal. Hence, deriving (simple) general investment recommendations is usually not possible.

Moreover, the overall capability of an information system to withstand attacks typically does not follow directly from the effectiveness of individual measures against particular attacks, because attackers may exploit any combination of vulnerabilities and potential attacks. Rather than identifying and correcting particular technical vulnerabilities, a more integrated approach is therefore required to thoroughly analyze a system's overall capacity to withstand attacks. In our research we thus address the following key questions:

– How can the overall security of information systems under varying conditions be assessed?
– How can information infrastructure designs be optimized with respect to organization-specific threats?
– How can decision makers effectively find the "best" compromise between costs, risks, and benefits of security investments?
– How can a tool-supported security decision process facilitate collaborative problem solving and involve both security experts and management?

To this end, we introduce a knowledge-driven approach for security decision making that links high-level risk estimates with lower-level technical implementation decisions. In particular, we develop a simulation-optimization architecture and propose a collaborative decision process that fosters communication between security professionals and managerial decision makers. Our decision support system enables both groups of stakeholders to systematically analyze the threats posed by various attackers based on criteria that are meaningful to both of them. This should result in more informed security decisions, generate a better understanding of security problems, improve situational awareness, lead to improved organizational alignment of business and IT, and bridge the gap between strategic security investment and operational implementation decisions.

The approach and the prototypical implementation were developed in the course of a four-year research project in a collaboration between researchers from three universities and a private-public partnership research center for information security. Preliminary results have been published in Kiesling et al (2013a,b, 2014) and Ekelhart et al (2015).

The remainder of the paper is structured as follows: Section 2 outlines several strands of related work and highlights the distinct contributions of our approach. Section 3 then describes a managerial process that facilitates sound security investment decisions based on our framework. Next, Section 4 introduces our knowledge model and the simulation-optimization architecture. Section 5 discusses implementation issues and Section 6 illustrates the methodology by means of an application example. We discuss feedback from an expert evaluation in Section 7 and conclude with an outlook on further research in Section 8.

2 Related work

Initially, the security of IT systems was largely perceived as a technical issue. As the information security discipline matured, the importance of a comprehensive approach to securing technology that includes processes, people, and other organizational factors was increasingly recognized (Baker and Wallace, 2007). This has led to an expanded research perspective that conceives security not as a matter of correcting individual technical vulnerabilities, but as an inevitable risk that has to be managed comprehensively. Following this line of reasoning, we focus our discussion of the extant literature on methods and tools that support *information security risk management* (ISRM), i.e., *"a process that allows IT managers to balance the operational and economic costs of protective measures and achieve* *gains in mission capability by protecting the IT systems and data that support their organizations' missions"* (Stoneburner et al, 2002).

In order to organize the rich ISRM literature and highlight contributions of our approach, we group existing approaches into methodological categories and illustrate differences in scope by means of a generic ISRM process model developed by Fenz and Ekelhart (2011). This model conceives ISRM as an iterative process that consists of five phases (see Fig. 1), namely, (i) system characterization, (ii) threat and vulnerability assessment, (iii) risk determination, (iv) control identification, and (v) control evaluation and implementation. The novel approach that will be introduced in Section 3 provides integrated tool support for all these ISRM phases.

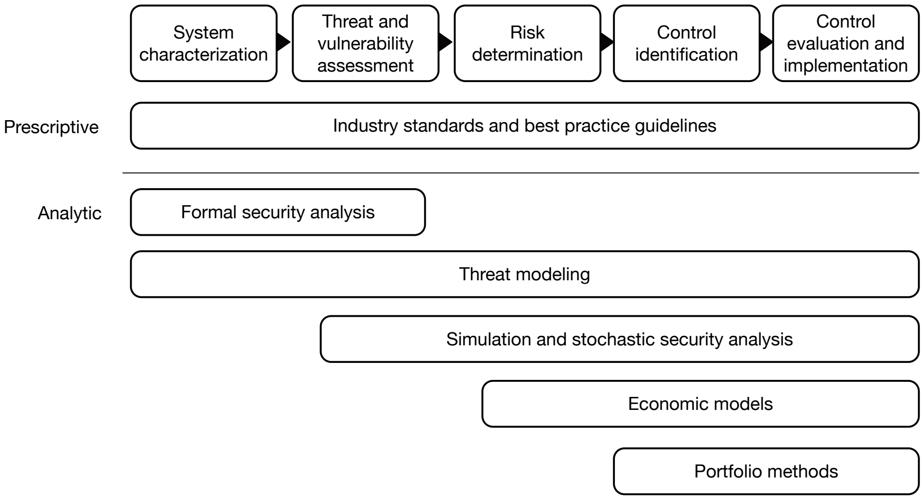

System characterization
Threat and vulnerability assessment
Risk determination
Control identification
Control evaluation and implementation
Prescriptive
Industry standards and best practice guidelines
Analytic
Formal security analysis
Threat modeling
Simulation and stochastic security analysis
Economic models
Portfolio methods

Fig. 1 Information security risk management phases covered by related approaches

Industry standards and best practice guidelines (e.g., NIST, 2011; BSI, 2013; ISO, 2013) provide instructional knowledge and guiding procedures for risk assessment and for the implementation of security management practices. Although awareness about the availability of these standards has been increasing in recent years, adoption among security practitioners is still relatively slow (Barlette and Fomin, 2010), particularly when compared to the diffusion of standards in other domains such as quality management (e.g., ISO 9001) or environmental management (e.g., ISO 14001) (Tunçalp, 2014).

Compliance to these standards should enable IT managers and technical personnel to better secure their IT systems. Whereas some of these prescriptive sources provide comprehensive knowledge about potential threats, vulnerabilities, and countermeasures (e.g., BSI, 2013), others focus mainly on the management process itself (e.g., ISO, 2013). Typically, they address all risk management phases in order to cover the complete risk management cycle. Some standards, however, only provide a rather generic description, and leave it to the organization to define its own risk management implementation. A detailed mapping of the individual elements of security standards as well as of methods in the generic ISRM process phases are discussed by Fenz and Ekelhart (2011).

The attacker-centric decision support system introduced in this work complements the defender-centric tool set used in such standards. Rather than providing general prescriptive guidance, our knowledge-based approach is based on comprehensive *system characterization* and *threat and vulnerability assessment* through modeling. Furthermore, the process integrates simulation-optimization to generate comprehensive organization- and threat-specific analyses for the *risk determination* and *control identification* phases. Finally, based on this input, we provide interactive decision support for *control evaluation and implementation.* This end-to-end process eliminates discontinuity between phases and allows security professionals and decision makers to retrace how results were derived.

Formal security analysis methods study the security properties of a modeled system by analytic techniques from the logic domain. Attack graph modeling, which formalizes a system as a finite state machine (Ammann et al, 2002; Ou et al, 2006; Sawilla and Ou, 2008), is an approach from this category that has attracted significant research interest. In such models, nodes represent states with respect to security properties and edges represent attack actions that trigger state transitions (Ma and Smith, 2013; Ritchey and Ammann, 2000). As in our model, attack actions may be specified as abstract patterns that provide a generic representation of a deliberate, malicious attack that commonly occurs in specific contexts (Moore, 2001). Other related work from this category includes a graph-based system model for vulnerability analysis that treats human and non-human "actors" fully symmetrically (Pieters, 2011).

Our approach is based on a description of attack patterns, which we enrich with additional aspects relevant for attack simulation and provide in a shared knowledge base (KB). However, motivated by the exponential search space that purely formal approaches (e.g., model checking) have to cope with, we use simulation rather than formal analysis. Despite efforts to reduce computational complexity (Ou et al, 2006), attack graph modeling is still impractical for large networks as complete enumeration of all possible attack paths is infeasible in such scenarios (Ma and Smith, 2013). Our solution tackles this issue by constructing an attack graph in a dynamic process that is driven by attackers' iterative step-by-step decisions.

Threat modeling methods provide a structured approach to identify threats and vulnerabilities in a target system. Attack tree modeling (Schneier, 2000; Mauw and Oostdijk, 2006) is a widely used method that falls within this category. It allows security analysts to hierarchically model different ways in which an attacker can achieve his or her goal. Similar to our simulation approach, attack trees are based on the idea of analyzing the security of a system from an attacker's perspective. The tree-based structure specifies an attack scenario in which the root node represents the attacker's goal, and paths from the leaf nodes to the root represent different ways of achieving this goal.

Several extensions and methods have been proposed for constructing attack trees efficiently and determining security metrics. Defense and protection trees incorporate security controls (cf. Bistarelli et al, 2006; Edge et al, 2006) and thereby extend the scope to the *control identification* and *control evaluation* phases. The influence of a security control can be analyzed by choosing comparable metrics for the protection tree. Bistarelli et al (2006) show how these results can be used to support the evaluation of IT security investments during the risk management process.

In practice, the modeling of attack trees is a labor-intensive manual ad-hoc process. Our approach therefore codifies reusable knowledge in a KB only once, and applies that security domain knowledge dynamically to system model instances. Furthermore, attack trees typically describe a particular attack scenario while our KB ideally covers the information system as a whole. Finally, we identify efficient security controls by running attack simulations.

Simulation and stochastic security analysis methods aim to discover attack paths in the system under evaluation, thereby covering the *threat assessment* and *risk determination* phases. By introducing countermeasures to the modeled system, they can also be extended to cover the *security control evaluation.*

Early work on attack simulation (e.g., Cohen, 1999) has largely neglected causal and temporal aspects. Later, Chi et al (2001) embedded a state-space model in a discrete-event framework as a means to capture simple causal mechanisms. This modeling approach is similar to ours, but its ability to handle non-trivial-sized problem instances is limited. Furthermore, it does not model adversary behavior and focuses exclusively on network security. A framework for the modeling and simulation of network attacks was developed by Franqueira et al (2009). It can be used to simulate an attacker who dynamically finds an attack path not through preconditions and postconditions, but by using an "access-to-effect" paradigm.

Other stochastic formalisms for dynamic security simulations proposed stochastic and interval-timed colored Petri Nets (Dahl and Wolthusen, 2006) and Generalized Stochastic Petri Nets (Dalton et al, 2006). In the latter paper, the authors aimed at automating the analysis of attack trees by using simulation tools, but they do not provide a complete framework for dynamic analysis.

Economic models have been used to derive quantitative risk estimates for a long time. In the security domain, Annual Loss Expectancy (ALE) (National Bureau of Standards, 1979) is one of the earliest and still most commonly used metrics to assess financial risks in the context of information systems. In this rather simple model, the expected loss due to a harmful event is the product of the estimated frequency of occurrence of an event and its estimated impact. The sum of the ALE of all events considered harmful results in an aggregate indicator for ALE. A major drawback of this approach is that it combines both risk factors into a single figure, which makes it impossible to distinguish high-frequency, low-impact events from low-frequency, high-impact events. Hoo (2000) extended ALE to account for security controls, which either lower the expected loss of a harmful event or its frequency. Using decision theory, it becomes possible to compare the performance of control sets, and hence, to support the *control evaluation* phase. Gordon and Loeb (2002) suggested cost-benefit analysis to evaluate information security investments. Other economic models, based on their respective roots in financial asset risk assessment, focus on return on investment (e.g., Mizzi, 2005) or value-at-risk (e.g., Jeevan and Rees, 2001; Wang et al, 2008). The latter category summarizes the worst conceivable loss due to a security breach over a target horizon with a given probability.

Economic models typically rely on expert estimates derived from prior experience, as well as from industry surveys, historical data, and so forth (e.g., concerning expected loss of a harmful event and frequency reduction of such an event achieved by applying a specific control). However, in practice it is very difficult to obtain the necessary data which is why the quality of estimates ultimately hinges upon subjective expert judgment. Moreover, experts usually do not base their assessment on the operational environment of a particular organization. Our approach aims to go beyond these high-level estimates in order to derive decision metrics from an executable system model through the explicit simulations of attacks which reflect the particular organization and its situation. The results obtained can then be traced and compared by studying the simulation output.

Portfolio methods support the *control identification* and *control evaluation and implementation* phases. They assist decision makers in the selection of efficient sets of security controls while accounting for interdependencies between controls within a given portfolio, often while accounting for multiple objectives. Wang et al (2006) and Islam and Wang (2008), for instance, optimize security controls based on attack trees models. In contrast to our work, their approaches focus exclusively on network hardening. Gupta et al (2006), as another example, use static attack trees as well, but do not account for attacker behavior. Moreover, the latter approach only considers the tradeoff between potential damages and costs, whereas our approach is more general with respect to objectives.

Earlier works by Strauss and Stummer (2002) and Neubauer et al (2006) also aim to support IT risk managers who search for the "best" individual portfolio of security controls with respect to multiple objectives. However, they differ in their approach of exploring the space of alternative solutions (i.e., portfolios). Both of them require means for a priori determination of the effects of a given portfolio of controls, but they rely on estimates rather than simulation techniques. The same holds for more recent work by Fenz et al (2011) which provides static risk analyses of an organization's infrastructure and then supports control evaluation through multi-objective decision support methods. To the best of our knowledge, none of the existing portfolio approaches in the security domain incorporate simulation-optimization and/or an explicit model of attacker behavior.

3 Decision support for security control selection

IT professionals responsible for securing information systems face several socio-technical and operational challenges. From an engineering perspective, the process of designing a system securely and selecting an appropriate set of security controls requires comprehensive analyses of the threat environment, the systems' vulnerabilities, and potential routes of attack. From a managerial perspective, on the other hand, security investment decisions are primarily motivated by the need to ensure compliance, avert losses, and balance risks. These differing security views and priorities can result in serious misalignment between stakeholder groups.

In order to reach consensus on their organization's security challenges and objectives, as well as to choose an adequate course of action, the problem has to be framed in terms that both groups – senior managers responsible for allocating resources and security managers responsible for implementing a chosen strategy

– can relate to. Our methodology therefore aims to provide an integrated framework for analyzing security issues and facilitating communication between senior management and security professionals. It structures the decision process around objectives that are meaningful to both groups of stakeholders (for a general discussion on the challenge of identifying the “right” objectives, cf. Keeney, 2013) and thereby links business concerns to operational implementation decisions, which may serve as a starting point for subsequent group decision and/or negotiation processes (for an overview cf. Vetschera, 2013). The decision support approach is structured as an iterative modeling and design process that is illustrated in Figure 2.

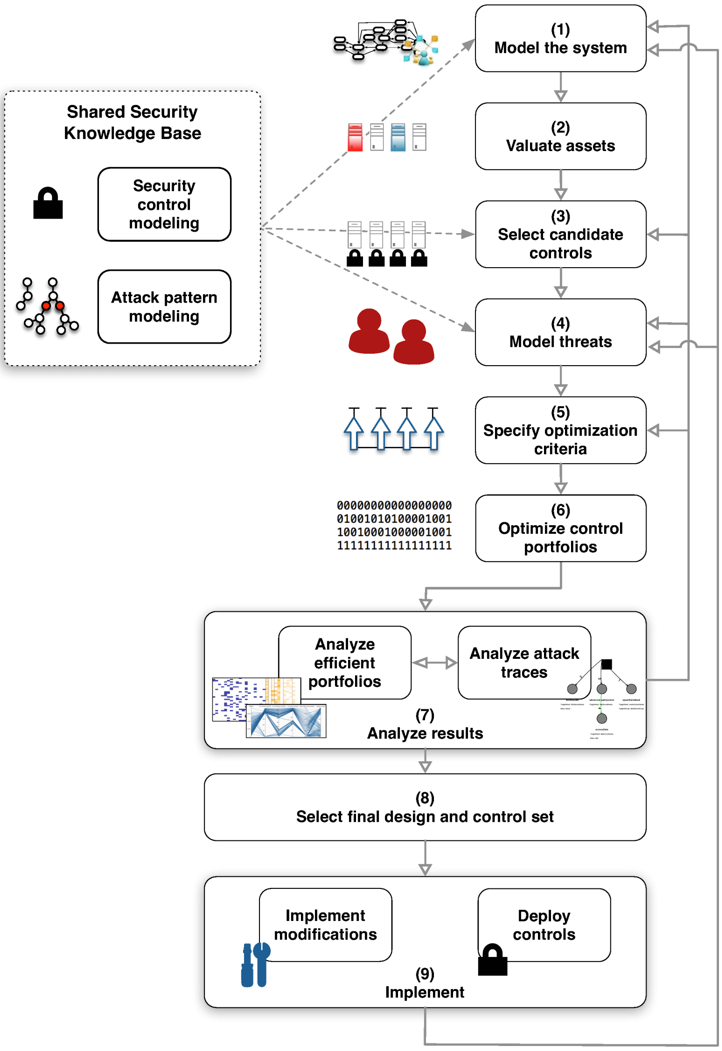


Fig. 2 Modeling and decision process overview

Step 1 consists in modeling the information infrastructure (e.g., hardware, software, data), its context (e.g., users, groups, access privileges, physical environment), and the controls that are already in place. Decision makers then valuate assets by assigning criticality ratings for security attributes such as confidentiality, integrity, and availability (Step 2) and select a subset of possible security control deployments to be considered in the subsequent optimization (Step 3). Each candidate control represents an assignment of a particular security control to a particular asset (e.g., installing a specific piece of software on a particular server). Next, decision makers model the threats that the system should be protected from (Step 4) and specify cost, risk, and benefit criteria for the optimization (Step 5, e.g., *minimize cost*, *minimize expected confidentiality impact*, *minimize undetected attacks*). In Step 6, a simulation-based optimization using metaheuristic solution algorithms is performed in order to identify efficient security control portfolios with respect to the specified objectives.

Subsequently, decision makers are supported in interactively exploring the solution space and can also conduct in-depth analyses of potential routes of attack for each candidate portfolio (Step 7). The process does not necessarily follow a strictly linear structure, i.e., at this point, the system model can be altered (i.e., return to Step 1), the set of candidate controls can be changed (i.e., go back to Step 3), alternative threat scenarios can be modeled (i.e., go to Step 4), and/or a different set of criteria can be chosen (i.e., go to Step 5). Once satisfactory results are obtained, decision makers select a final design and control set (Step 8), implement system design modifications, and deploy the selected security controls in the system (Step 9).

Due to the evolving nature of information systems and security knowledge, we recommended to establish this methodology as part of a continuous improvement process. The system model must therefore be updated regularly to reflect changes to the infrastructure (hardware, software, and configuration changes; provisioning and de-provisioning of user accounts, etc.). At present, such alterations to the system model must be made manually, but this process can be (partly) automated in the future. Moreover, it is necessary to adapt asset valuations to reflect changes in business criticality and to change the attacker model as the threat landscape changes. Furthermore, security knowledge must be updated to account for new vulnerabilities and potential attack vectors. To this end, it is possible to share information on attacks and the effectiveness of controls in a common security KB (illustrated on the left-hand side in Fig. 2). This repository could be provided as a service that users can subscribe to in order to cope with an evolving threat environment. Through an automated process, decision makers would obtain periodic updates, could directly re-evaluate the security of their systems, and re-optimize them once new types of attacks become known and new security controls are available.

4 Simulation-optimization architecture

Figure 3 provides a high-level overview of the MOSES^3 architecture for simulation-driven security control portfolio optimization.[1]

[1] The acronym MOSES^3 stands for $\underline{\text{M}}$ulti-$\underline{\text{O}}$bjective decision $\underline{\text{S}}$upport in $\underline{\text{E}}$fficient $\underline{\text{S}}$ecurity $\underline{\text{S}}$afeguard $\underline{\text{S}}$election.

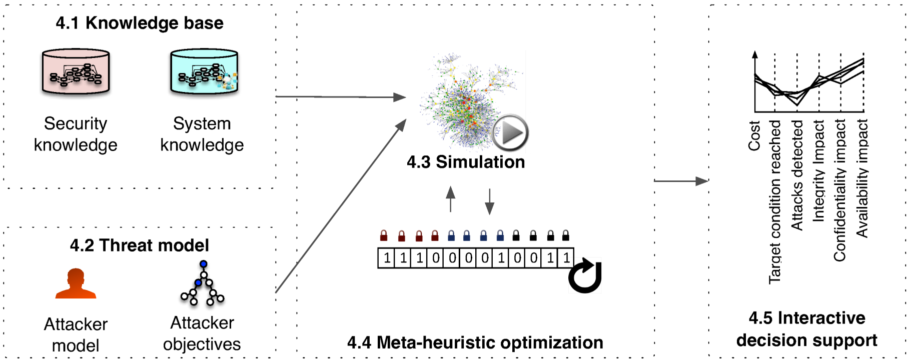


Fig. 3 Architectural overview

The attack simulation engine is at the core of this architecture. It relies on a comprehensive KB that captures information on the information system to be protected, viable attack mechanisms, and the effect of security controls (see Sect. 4.1). Furthermore, the simulation makes use of a threat model that consists of (i) an attacker model (i.e., a formal characterization of attackers, including their skills, resources, behavior, and points of access); and (ii) an optional specification of attacker objectives such as obtaining access to a particular data set (see Sect. 4.2). Threat models that do not use specific attacker objectives can be based on the assumption that attackers strive to maximize the impact of their attacks without aiming for a particular target.

Individual attack patterns are linked dynamically during the simulation of attacks on a given system configuration. Such a system configuration is encoded as a genotype string that specifies which security controls are deployed on a particular asset. For each system configuration under consideration, a number of simulation runs are performed and the aggregate outcomes (average impact caused by the attacker, detection rate of attacks, share of successful attacks, etc.) are recorded (see Sect. 4.3). These outcome measures are used as optimization criteria in the search for efficient configurations. Because the number of potential configurations grows exponentially with the number of candidate controls and asset assignments, complete enumeration of all potential configurations of a large IT infrastructure is usually computationally intractable. We therefore use a genetic algorithm that provides an approximation of the set of Pareto-efficient security control portfolios within reasonable runtime in most instances (see Sect. 4.4).

Finally, several alternative user interfaces support decision makers in the interactive exploration of the identified solution space and the selection of a preferred system configuration (see Sect. 4.5). The remainder of this section discusses each of these components in detail.

4.1 Knowledge base

The KB is divided into a security KB and a system KB in order to separate general attack and defense knowledge from the organization-specific system knowledge. This partitioning into abstract security and concrete system knowledge facilitates the sharing and reuse of domain-specific attack knowledge so that organizations only need to model their specific environment. The security KB, which can be reused by other organizations, formally describes attack patterns, i.e., individual mechanisms of attack, their preconditions, and effects upon execution. It also contains information on how preventive and detective security controls interact with these patterns. The system KB, on the other hand, captures information on IT infrastructure elements and their relation to each other; this includes a wide range of entities, such as users, user groups, computers, services, and network devices. Both, the security KB and the system KB are implemented in the logic-oriented programming language Prolog.

4.1.1 Security knowledge base

The security KB contains attack patterns and security control definitions. Attack patterns describe formally how systems can be compromised by an attacker; control definitions specify how assets can be protected by applying security controls. Each attack pattern is applicable under a set of preconditions (e.g., an action can require a certain system state or particular attacker skills). Furthermore, an attack pattern defines postconditions to specify the effect of the action upon execution.

Attack patterns used in this paper are taken from the publicly available CAPEC (Common Attack Pattern Enumeration and Classification) repository,[2] which currently includes more than 400 patterns. These patterns are described in a semi-structured manner and need to be formalized before they can be used within the simulation. To this end, we derive preconditions and postconditions from the information provided in the CAPEC sections on *Attack Prerequisites*, *Experiments* (attack steps), *Outcomes* (success and failure results), and *Summary.*

Postconditions define the outcome of an attack action (i.e., the state transition after an attack action has occurred). At simulation runtime, each executed attack action will either succeed or fail. Upon a successful attempt, an attacker may, for instance, gain access to the attacked system whereas a failure may make the target system inaccessible. Each outcome type (*success*, *fail*) triggers a respective rule that alters the current system state in the system KB. The probability of success for each action is derived from the *Typical Likelihood of Exploit* section inside the CAPEC description.

The CAPEC database also provides information on the *impact* of an attack by referring to the security attributes *confidentiality*, *integrity*, and *availability* (CIA) on a scale of *low*, *medium*, and *high.* Though this is relevant information, it cannot be assigned to actions directly for our purposes because impact is highly context-dependent. Therefore, the simulation model does not determine impact metrics solely based on the fact that an attack action has occurred, but also accounts for the valuation of the asset being attacked. To this end, every attack pattern defines the affected security attributes. A specific attack, for instance, may only impact the target's availability, whereas another attack may impact confidentiality, integrity, and availability of the target asset.

Finally, we use CAPEC's *Mitigations*, *Solutions*, and *Relevant Security Requirements* sections to enrich the KB with security control information. We distinguish between detective controls (which detect attack actions) and preventive

[2] `http://capec.mitre.org/`

security controls (which may inhibit or hamper attacks) and model the controls accordingly.

4.1.2 System knowledge base

The information system to be protected, its constituent components, and the relationships between them are modeled in a system KB. We use a broad definition of the term "asset" for all these elements, including tangible assets such as computers, servers, rooms, and employees, and intangible assets, such as data, reputation, and policies.

In order to register the impact of attacks during the simulation, assets are assigned criticality ratings based on existing asset criticality reports or impact analyses. These ratings can represent monetary values or use other quantitative or qualitative scales (e.g., 3 stands for a *high* level of criticality, 2 for a *medium* level, and 1 for a *low* level).

4.2 Threat model

Attackers are heterogeneous in their motivations, resources, capabilities, and points of access. In order to analyze the threats that they pose, and to identify appropriate controls, it is necessary to explicate assumptions about the threat agents. Whereas attackers are usually classified based on a natural language description (e.g., external, internal, government, secret services; cf. Panchenko and Pimenidis, 2006), we take advantage of our formal model to define more specific attacker profiles. These profiles include specifications of attackers' capabilities, resources (e.g., time, equipment), risk preferences, and other behavioral attributes (e.g., propensity to alternate between different attack strategies). Threat models determine whether particular attack patterns are available, which in turn affects the choice of attack paths in the simulation. Moreover, initial access privileges can be specified where appropriate in order to reflect, for instance, that an employee has a more comprehensive set of access permissions than an "outsider".

We associate attacker profiles with a behavioral model that iteratively selects attack actions based on: (i) individual attacker characteristics; (ii) the attacker's general knowledge about possible routes of attack; and (iii) the outcomes of prior attack actions. In the following, we motivate two such exemplary models.

Goal-driven behavior The first behavioral model is applicable for attackers who aim for a particular goal that can be expressed as a target condition (such as *access* to a particular data asset). In this case, it is assumed that attackers possess general security knowledge, but they do not have complete and concrete information about the system that they are going to attack. We use an abstract attack graph to represent general knowledge about possible routes of attack and let attackers use it as a mental map of how actions can be combined to achieve a particular outcome. At the beginning of an attack simulation, this graph is constructed for a particular attacker and target condition by querying the security KB. In the course of the simulation, the KB is then queried for valid asset assignments for all preconditions on abstract attack actions whenever an attacker has to make a decision on how to proceed. This provides a set of action instances (i.e., abstract actions with assigned variables) that can be executed against particular assets.

Out of this set of potential actions, only a subset is actually considered each time a decision is made. This “choice set” is based on the notion that attackers alternate between following a chosen attack path and trying new approaches. Accordingly, attackers can choose newly available actions after a successful attack action, retry a failed action, choose an action that is similar to a previous one, or choose a completely new attack vector. The likelihood of each of those behaviors is controlled through probability functions.

In modeling the attacker’s decision process for selecting the next attack action from the choice set, the following assumptions are made. First, we assume that attackers minimize their expected effort and hence they tend to launch attacks “close” to the target condition (i.e., prefer actions for which the expected number of steps required to achieve their objective is smaller). Second, we stipulate that attackers prefer actions that have a higher probability of success. Finally, we assume that attackers prefer actions with a lower probability of detection (i.e., they aim to avoid being recognized). The relative importance that attackers place on these aspects can be controlled through preference weights.

Reward-driven behavior The second behavioral model represents attackers who do not aim for a particular outcome, but rather strive to maximize the impact of their attacks. This model is suitable for opportunistic threat profiles and matches in cases in which attackers act impulsively, i.e., are driven by immediate rewards rather than by a strategic goal. In this model, attackers select attack actions with weights proportional to the impact they may cause. The impact attackers strive for can be defined in terms of security attributes (such as confidentiality, integrity, and availability) or in more quantitative (e.g., financial) terms. Furthermore, the impact from an attacker’s perspective (e.g., in terms of financial payoff or satisfaction obtained from a successful “hack”) does not necessarily equal the operational impact from the organization’s perspective. In order to simulate attacker behavior adequately, additional impact attributes can be used to assign a payoff of a successful exploit for an attacker to individual assets.

4.3 Simulation

The attack simulation accesses information on attack patterns, controls, and the IT system infrastructure from the security and system KBs and executes attacks for given threat scenarios. Attackers’ choices regarding their course of action, the outcome of individual attack actions, and the detection of attacks are determined probabilistically in the simulation. It is therefore necessary to perform multiple simulation replications using different random seeds to tackle uncertainty and variability. For each replication, the simulation executes a schedule of discrete events and records outcomes for further analyses and aggregation. This approach is capable of capturing complex causal relationships and timing interactions. Figure 4 illustrates the types of events used in the simulation and how they are scheduled.

The beginning of each attack action is represented by an *action start event*. Upon execution, instances of this event type determine the effective duration of the attack action and an *action end event* is scheduled accordingly. The actual

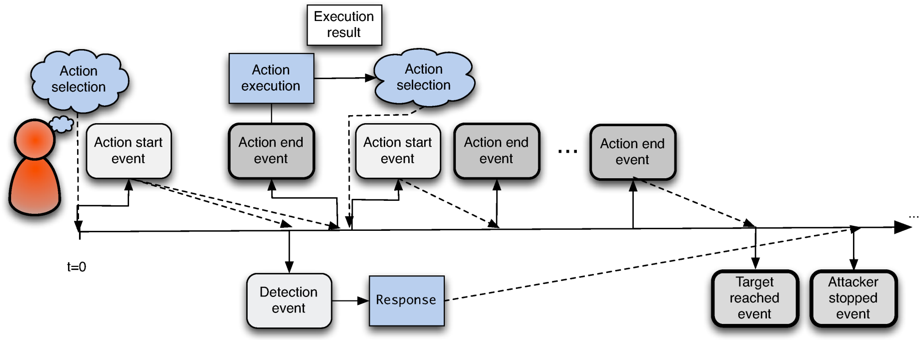


Fig. 4 Event types and scheduling of events

duration depends on factors such as difficulty, attacker skills, preventive controls applied on the asset under attack, and randomness due to inherent variability. In case detective controls are associated with an asset that is being attacked, *detection events* may be scheduled immediately or with a random delay, as specified in the control model. If the attack target is reached after the current action has been executed, a *target condition reached event* is scheduled immediately. Detective events may also completely terminate an attack by scheduling an *attacker stopped event*.

During the simulation runs, each attack action performed by an attacker is immediately evaluated to determine (i) whether the action was successful; (ii) whether the target condition has been satisfied; (iii) which new actions are available after applying the action's postconditions; (iv) which actions become unavailable due to the action's outcome; and (v) the impact on security attributes (e.g., on *confidentiality*, *integrity*, *availability*). The impact severity of an executed action is determined from the system KB by querying the corresponding security attribute's asset valuation and, thus, depends on the execution context.

Controls can either stop the attacker, which terminates the simulation, or alter the system state as specified by a postcondition (a server crash, for instance, can affect the asset's availability property).

4.4 Meta-heuristic optimization

For optimization purposes, each portfolio is represented by a genotype, i.e., a string of binary variables that indicates which security controls are assigned to a particular asset (e.g., one digit in the string may indicate whether *antiVirusSoftware1* is deployed on *workstationHosts*). Upon evaluation of a portfolio, the system model is initialized according to the genotype of the portfolio under consideration and then attacks are simulated with varying random seeds. The outcomes of the simulated attacks are monitored and aggregated across replications (e.g., using *min*, *max*, *average*, *median*, *sum* as aggregation functions). Since multiple objectives are taken into account, there is typically no final single "best" solution. Rather, the optimization usually results in a set of non-dominated portfolios that are Pareto-efficient insofar as no other feasible portfolio exists that achieves at least equal values in all objectives and a strictly better value in at least one objective.

The simulation-optimization problem for identifying Pareto-efficient portfolios is highly challenging computation-wise because of the large combinatorial decision space (with 2^n potential portfolios, where n is the number of control-asset combinations), the expensive simulation-based evaluation procedure, and the need to account for multiple optimization criteria. Exact solutions (i.e., complete sets of all efficient portfolios for a given number of simulation replications) can only be determined through complete enumeration for rather small problem instances. In order to tackle larger problem instances we considered several metaheuristic solution procedures (e.g., tabu search, variable neighborhood search, simulated annealing, ant colony optimization) and finally opted for genetic algorithms. They evolve a population of individuals (control portfolios) by evaluating their fitness (assessing criteria), selecting fit individuals (control portfolios), and performing crossover and mutation operations on the selected individuals in order to generate offspring. The binary genotype strings generated in this process are evaluated by means of multiple simulation replications, which yield aggregate criterion values that then are used for assessing the "fitness" of the respective control portfolio.

We ran experiments with two variants of multi-objective genetic algorithms – namely, SPEA2 (Zitzler et al, 2002) and NSGA-II (Deb et al, 2000) – for several smaller problems for which it was still possible to enumerate the complete search space in reasonable time. It turned out that both genetic algorithms performed reasonably well, with NSGA-II yielding slightly better results (for more details cf. Kiesling et al, 2013b). Even with default optimization parameters, the NSGA-II procedure already identified roughly 65% of all efficient portfolios within just 20% of the runtime for the complete enumeration. Its performance could be further improved by fine-tuning parameters (e.g., the mutation rate) or by adapting genetic operators (e.g., using a two-point rather than a one-point crossover). More substantial measures (e.g., seeding, exploiting domain knowledge about the genotype structure, caching, statistical analysis of simulation results and, accordingly, adapting the number of simulation replications, parallelization, or using surrogate models) are outlined by Kiesling et al (2015).

4.5 Interactive decision support

Portfolio selection problems with multiple evaluation criteria and a large number of alternatives place a significant cognitive burden on the decision maker. Various approaches for alleviating that problem have been proposed in the decision support literature. They can be broadly categorized into three groups (Vincke, 1992). Approaches from the first group aim to elicit (all) the decision maker's preferences, express them in an explicit function (e.g., by using multiple attribute utility theory), and accordingly calculate the single "best" solution that provides the highest utility value. Outranking approaches (e.g., ELECTRE) form the second group. They aim to capture "just" the decision maker's strongly established preferences, outrank portfolio alternatives with respect to these strong preferences, and for the remaining solution set, exploit further outranking relations. The third group comprises interactive methods that require even less *a priori* preference information. They allow the decision maker to gradually learn about his or her implicit preferences in the course of an interactive procedure. To this end, efficient solutions are calculated and the decision maker is supported in exploring the solution space (for an example cf. Stummer et al, 2009).

With respect to the portfolio selection problem at hand, we found that IT managers typically cannot (or are not willing to) provide sufficient preference information in advance, because the tradeoffs involved in security decisions are often not apparent to them and, moreover, such tradeoffs can differ considerably between scenarios. Accordingly, we follow the paradigm of the third group of approaches and provide three alternative types of visualizations of the multidimensional data obtained from the optimization: parallel coordinates, radar charts, and heatmaps. The parallel coordinate visualization also offers interactive mechanisms for selecting subsets of solutions by graphically imposing upper and lower bounds on the criteria (for a more general comparison of parallel coordinates and heatmaps in portfolio selection cf. Gettinger et al, 2013).

Parallel coordinates lay out a set of axes in parallel (for a detailed description cf. Inselberg, 2009). In the portfolio selection context, we use a separate axis for each criterion and represent each portfolio by a profile line that intersects each axis according to the value achieved in the respective criterion (for illustrative examples in the context of our application case being presented in Sect. 6, see Fig. 6 and Fig. 7). The axes can be rearranged via drag and drop, which allows the user to identify patterns such as positive and negative correlations. Upper or lower bounds for objectives can be imposed by dragging bars to mark admissible intervals. During dragging operations, the decision support system indicates portfolios that will be eliminated as a result of an additional restriction by graying them out (and displaying them in regular color once intervals are widened again).

This visualization exhibits geometric interpretability and provides an excellent overview of the distribution of values. It does not show the genotypes of the portfolios directly, but rather abstracts the problem from the underlying design considerations. Hence, parallel coordinates are particularly suitable for exploring the criteria space, focusing entirely on solutions and tradeoffs rather than questioning which individual controls to implement.

Radar charts lay out variables concentrically on equiangular axes originating at the same point (for an overview cf. Draper et al, 2009, for illustrative examples in the context of our application, see Fig. 8). Radar charts provide the most detailed view and are particularly suitable for comparing individual portfolios after the solution space has been narrowed down to a small number of alternatives.

Heatmaps are essentially matrices in which the cells are colored according to the cell value; they have been adopted from visualization methods developed in data mining (Cook et al, 2007; Lotov and Miettinen, 2008). We use heatmaps to provide a highly condensed overview of the design and the criteria spaces simultaneously (an example is provided in Fig. 9). Each line represents an efficient solution. On the left-hand side of the heatmap the design of a portfolio is visualized by using blue squares for controls that are included in a particular portfolio and white squares otherwise. The performance of a portfolio with respect to the criteria being considered is depicted on the right-hand side of the heatmap.

A key advantage of the heatmap visualization lies in the fact that it provides a good overview of the frequency of controls in the efficient portfolios. Furthermore, heatmaps can reveal patterns such as correlations and tradeoffs between criteria.

5 Implementation

Most parts of the prototypical implementation of our MOSES[3] architecture for simulation-driven security control portfolio optimization were coded in Java. We used the scheduling mechanisms provided by MASON (Luke et al, 2005), a fast discrete-event simulation core which also provides the pseudo-random numbers used in the simulation. The genetic algorithm-based optimization was implemented using the metaheuristic framework Opt4J (Lukasiewycz et al, 2011).

The KB was written in SWI-Prolog (Wielemaker et al, 2012) and accessed in Java via JPL[3]. In an earlier prototype, we experimented with an OWL[4] KB and reasoner and used the query language SPARQL[5] to obtain available actions in each simulation step. OWL provides rich expressiveness and the Protege API provided the interface for assertions in the KB. It turned out, however, that the queries became exceedingly large and complex, and ultimately, were impractical for optimization purposes due to performance limitations. Prolog queries for finding possible attack actions, in comparison, are very efficient. Another advantage of Prolog in this context is that query parts can be reused easily through the combination of multiple rules. Finally, our decision support tool, the interactive parallel coordinates, and the radar chart visualization are implemented using D3.js[6], a JavaScript library for document manipulation.

6 Application example

We demonstrate the applicability of the proposed methodology by means of two scenarios. In the first scenario, decision makers are concerned with protecting a particular asset against attacks from specific adversarial types. For this scenario, we conducted independent optimization experiments for each attacker profile. In the second scenario, decision makers are generally interested in protecting their infrastructure against external and internal attackers. For this scenario, we simultaneously optimized the system against multiple attacker types. Both scenarios use a shared KB outlined in the following.

6.1 Knowledge base

As to the required *security knowledge* for our application, we first modeled a set of example CAPEC attack patterns and added them as formally specified actions, complete with all necessary pre- and postconditions. Where appropriate, we created multiple attack action instances from a single CAPEC pattern to reflect the

[3] http://www.swi-prolog.org/packages/jpl/
[4] http://www.w3.org/TR/owl2-overview/
[5] http://www.w3.org/TR/sparql11-query/
[6] http://d3js.org/

fact that these attacks can be performed at multiple levels of sophistication. Each instance was assigned a skill level that determines the types of attackers they are available to. A basic SQL injection, for example, can be executed by low-skilled attackers by simply pasting code snippets available on the web into forms on a website. An advanced SQL injection, however, requires extensive knowledge and significant skills to create custom exploits for the particular system being attacked. Basic and advanced attack actions differ in their respective success probabilities (more advanced attackers will typically have a greater likelihood of successfully executing an attack action) and the effectiveness of countermeasures (countermeasures are typically effective against attacks executed by a low-skilled attacker, but less so if the same type of attack is carried out by an advanced attacker).

Next, we added a *zero-day* attack action that exploits previously unknown vulnerabilities and complemented the attack actions with reconnaissance actions that provide attackers with general information required to commence an attack (e.g., *scanNetwork* is required to discover potential target hosts for an attack). Furthermore, we modeled legitimate actions (e.g., *accessData*) that attackers can use maliciously once the necessary preconditions have been fulfilled (e.g., the required credentials have been obtained). Table 1 provides an overview of the resulting attack actions used, their respective CAPEC reference numbers, and their skill level requirements (see Tab. 2 in Sect. 6.3 for a specification of skill levels assigned to the individual attacker profiles).

Table 1 Attack patterns used in application case

Attack action	CAPEC (ID)	Skill level
Buffer overflow	Buffer overflow in an API call (8)	2
Buffer overflow advanced	Buffer overflow in an API call (8)	3
Brute force	Password brute forcing (49)	0
SQL injection	SQL injection (66)	1
SQL injection advanced	SQL injection (66)	2
Email keylogger	Email injection (134)	1-2
Email keylogger advanced	Email injection (134)	3
Email backdoor	Email injection (134)	1-2
Email backdoor	Email injection (134)	3
Directory traversal	Directory traversal (213)	>1
Zero day	Privilege escalation (233)	3
Scan network	Port scanning (300)	1-2
Scan network stealth	Port scanning (300)	3
Shoulder surfing	Social information gathering (404)	0
Spearfish attack	Social information gathering via pretexting (407)	2
Social attack	Information elicitation via social engineering (410)	<3
Access data	Legitimate action	0
Access host	Legitimate action	0

Finally, we enriched the security KB with security control definitions that were also derived from CAPEC patterns. Security controls included, for instance, anti-virus software, software patches, intrusion detection systems, log policies, and security trainings.

As to the *system knowledge* base, the model of the example organization's IT infrastructure was generated automatically by first randomly creating instances of

Host, *Subnet*, *Data*, and *User* and subsequently adding relationships between them (e.g., *Host stores Data*, *Host uses Software*, *User in UserGroup*). The synthetic system used in this example consisted of 30 hosts, 5 web servers, 5 database servers, 30 employees, and 3 administrators (see Fig. 5 for an overview; details such as connections between systems and installed software have been omitted for the sake of clarity).

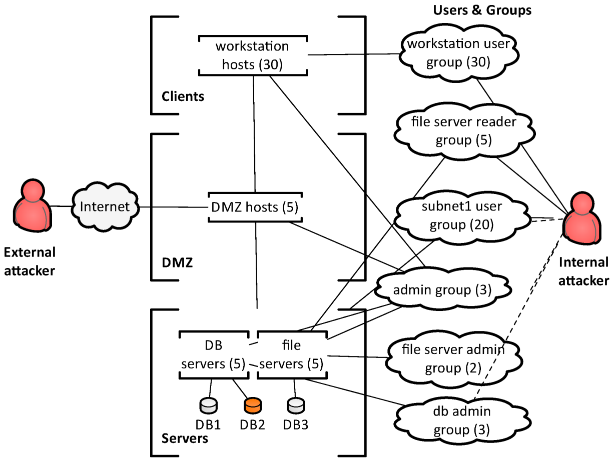


Fig. 5 Scenario system model overview

6.2 Scenario 1: Targeted attacks

The first scenario aims at protecting data set *DB2* which holds highly sensitive information. The simulated attackers only stop their attack once they have reached the target condition or they run out of time. In order to analyze the effectiveness of control portfolios for particular attacker types, we optimized the IT infrastructure independently for each of the attacker profiles summarized in Table 2. In our model, internal attackers already have the means to access a system and can therefore follow more direct attack paths to reach the target condition. Shoulder surfing (i.e., obtaining a password by watching a person typing it) may serve as an example. External attackers, by comparison, have more limited means and must, for instance, first find an entry point to the organization's internal network through hosts from the "demilitarized zone" (DMZ) before they can launch attacks on internal hosts.

We used the following six objectives in our simulation-optimization experiments, all of which had to be minimized: implementation cost of the controls, number of undetected attacks, number of attacks for which the target condition has been reached, and impact of attacks on confidentiality, integrity, and availability, respectively. For the latter three criteria we use a lexicographical scale and map the impact category valuations *low*, *medium*, and *high* to a scalar criterion value such that ultimately only the highest impact category is of relevance (i.e.,

Table 2 Attacker profiles in Scenario 1

Attacker	Time (sec.)	Skill level	Initial access
Internal	150,000	0	Workstation hosts, Fileserver hosts
Skilled internal	150,000	2	Workstation hosts, Fileserver hosts
External	200,000	1	Attack client
Skilled external	150,000	2	Attack client
Advanced persistent threat	1,000,000	3	Attack client

a single medium impact on confidentiality is always considered more critical than any number of low impacts).

The NSGA-II optimization ran for 500 generations with the default parameters suggested in Deb et al (2000). The size of the archive of proposed efficient solutions kept in each generation was unlimited. For each evaluated portfolio, 50 simulation replications with varying random seeds were performed and the objective values were averaged across replications to determine the portfolio's objective values. Typically, the search procedure converged within the first 250 generations and the set of proposed Pareto-efficient portfolios remained stable thereafter. The optimization runs for each attacker profile were executed on single nodes of a scientific cluster each equipped with a 2.66 GHz Intel Xeon processor and took between 9.5 hours (*skilled external* profile) and 48.7 hours (*employee* profile) to complete.

The number of proposed efficient control portfolios varied significantly across attacker profiles; the optimization resulted in 341 efficient control portfolios for *advanced persistent threat* attackers, 92 for *skilled external* attackers, 57 for *script kiddie*, 242 for *skilled internal*, and 94 for *unskilled internal* attackers. Without any controls deployed, *skilled external* attackers, for example, on average reached the target condition in 76% of the simulation runs, which could be reduced to just 18% when the most effective control portfolio is in place. Detection rates for attacks from this group of attackers vary between 0% and 94% depending on the control portfolio deployed. *Unskilled internal* attackers, by comparison, on average obtain access to the target data set in 38% of the simulation runs if no controls are in place. It is noteworthy that even with the most effective portfolio this share cannot be reduced below 28% (which is still considerably high). This result can be explained by the internal attackers' capability to take advantage of social attacks and their privileged access to internal systems, which makes most controls ineffective in preventing successful attacks. Detection rates, however, reach 92% for the internal attacker with proper detective controls (e.g., effective security training) in place.

Decision makers can analyze the optimization results obtained for each attacker profile and thereby obtain a deeper understanding of possible attack paths. Furthermore, they are supported in the exploration of various alternative (efficient) control portfolios to secure the system. In the following, we exemplify such an exploration process for the *skilled external* attacker. Figure 6 provides the objective values for all efficient portfolios in an interactive coordinate plot. By setting upper and lower bounds for any objective, a decision maker can restrict the admissible solution range and, thus, iteratively narrow down the portfolios that fulfill his or her requirements. A decision maker interested in a portfolio with low costs, for instance, can interactively filter portfolios with costs of 6,000 or less by dragging (i.e., lowering) the upper limit indicator on the cost axis. Furthermore, we suppose that the decision maker in our example strives for a detection rate of at least 50% and is only interested in solutions in which less than 30% of the attacks reached the target condition. This reduces the set of portfolios that fulfill all requirements to six, as shown in the screen capture of the user interface in Figure 7. To allow the user of our decision support system to more easily identify patterns such as positive and negative correlations among criteria, the axes can be rearranged by simply dragging them. In Figure 7, this was done with axes *Target condition reached* and *Undetected* that were moved to the left. Objective values of the remaining control portfolios are listed in Table 3.

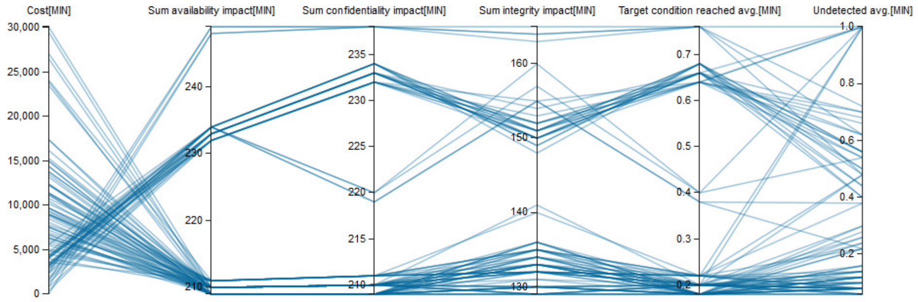


Fig. 6 Criterion values for skilled external attackers in Scenario 1

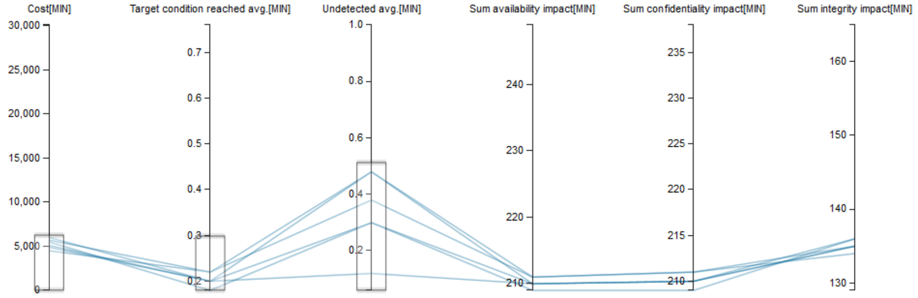


Fig. 7 Criterion values for skilled external attackers after restrictions in Scenario 1

Decision makers can continue to interactively explore the solution space by imposing and relaxing constraints. Furthermore, they can select two or more portfolios and compare them in a radar chart for a more detailed visualization. This is demonstrated in Figure 8 for Portfolio 1 (colored orange) and Portfolio 6 (colored blue). Since all the criteria considered have to be minimized, the axes in each category are scaled from the worst (i.e., highest) value achieved by one of the (in this case: 92) Pareto-efficient portfolios in the center of the chart to the best (i.e., lowest) value at the outside of the chart. In other words, the farther away

Table 3 Remaining control portfolios after restrictions in Scenario 1

Id	Cost	Target reached	Undetected	Availability impact	Confidentiality impact	Integrity impact
1	4,500	0.22	0.48	211	211	135
2	4,900	0.20	0.48	210	210	136
3	5,100	0.20	0.30	210	210	135
4	5,500	0.18	0.30	209	209	136
5	5,700	0.22	0.38	211	211	134
6	6,000	0.20	0.12	210	210	135

from the center the better it is. Note that the two portfolios in our example have already been filtered with respect to costs, detection rate, and target reached (as described above) and therefore look rather similar. Still, it can be seen that a considerably higher detection probability as well as a slightly better protection (e.g., a lower percentage of attacks that reached the target condition) can be achieved by implementing Portfolio 6. On the other hand, Portfolio 6 is also somewhat more costly.

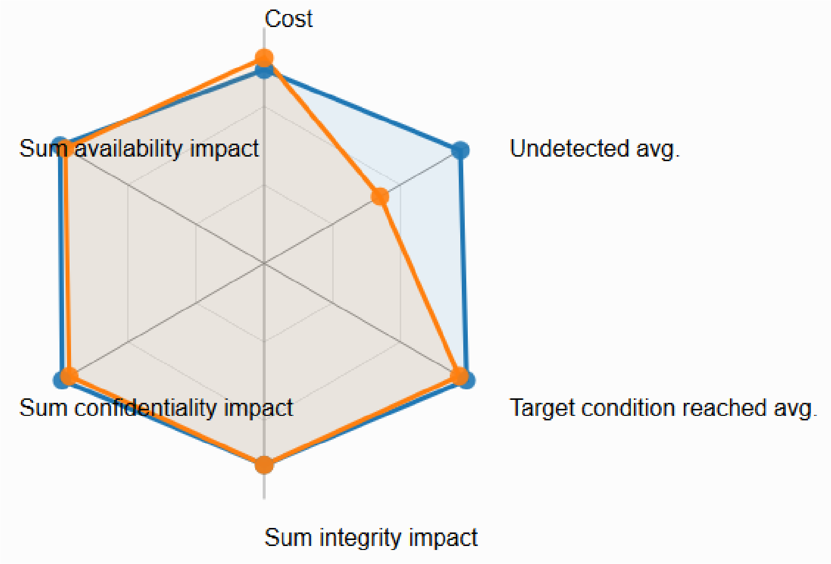
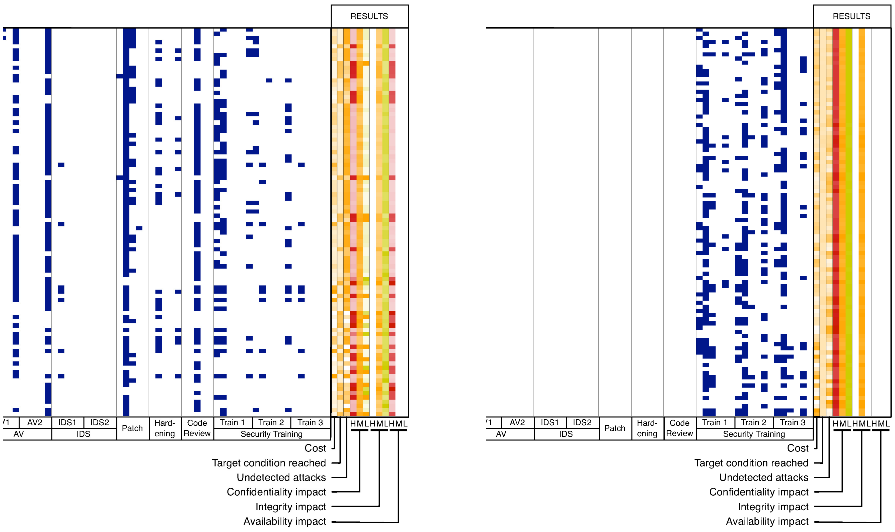


Fig. 8 Radar chart visualization of two portfolios in Scenario 1

Whereas the radar chart visualization facilitates a detailed analysis of strengths and weaknesses of a few selected portfolios, heatmaps provide the decision makers with a more general overview. They are particularly useful when comparing the relative threats posed by the different types of attackers and the effective controls to counteract them. Heatmaps visualize the performance of each portfolio – encoded through colors ranging from green (low costs, high detection rate, etc.) to red – on the right-hand side and, in our implementation, they provide information on the security controls that are included in the respective portfolios on the left-hand side. Figure 9 provides an exemplary comparison of *skilled external* and *unskilled internal* attackers. From the right heatmap (Fig. 9b), for example, it can be learnt that none of the efficient security control portfolios that are supposed to protect (only) against internal attackers include technical controls. Instead, they focus on social controls, which is reasonable since internal attackers use various social attack techniques. Also, it can be seen that internal attackers always cause high confidentiality impacts, irrespective of the applied control portfolios.



(a) Skilled External

(b) Internal (Employee)

Fig. 9 Heatmap comparison of skilled external attackers with internal attackers

6.3 Scenario 2: Multiple attackers

For the second scenario, we defined a *skilled external* and a *skilled internal* attacker profile and optimized the system against both types of attackers simultaneously (the parameters used are listed in Tab. 4). The modeled attackers strive to maximize the confidentiality impacts caused and continue their attacks until they either run out of time or attack actions (see Sect. 4.2 for details on the behavioral model). The minimization objectives used for the optimization were costs for the implementation of the controls, the number of undetected attacks, and the confidentiality impact. Once again, we performed 50 simulation replications for each portfolio. Because this optimization is more difficult for multiple attackers, we ran the genetic algorithm for 1000 generations to ensure for convergence. The multiple optimization runs took 12.6 hours to complete.

In total, 171 efficient portfolios were identified for Scenario 2. Figure 10 depicts the parallel coordinates representation of these portfolios, for which the decision makers can interactively set aspiration intervals. Figure 11 provides the heatmap

Table 4 Attacker profiles in Scenario 2

Attacker	Time (sec.)	Skill level	Access
Skilled external	200,000	2	Attack client
Skilled internal	200,000	2	Workstation hosts, Fileserver hosts

representation for a high-level overview. The latter shows that some of the proposed efficient portfolios focus on security training and thus prove effective against internal attackers, whereas other portfolios are mainly composed of technical controls and are, therefore, particularly effective against external attackers. Most of the portfolios, however, combine various control types, which hardens the IT infrastructure under consideration against both types of attackers.

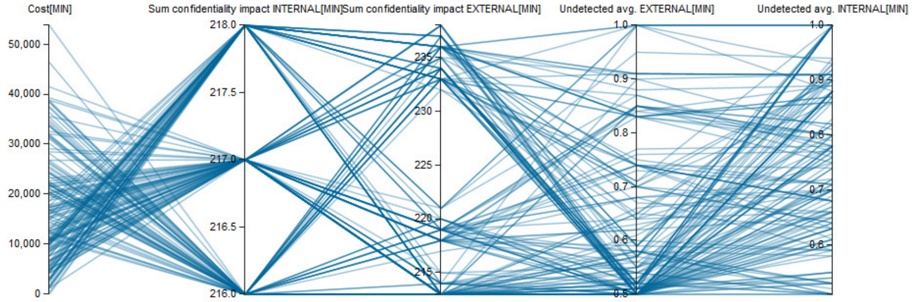


Fig. 10 Objective values for multiple attackers in Scenario 2

7 Expert evaluation

In order to evaluate our methodology and gather further ideas for improvement, we conducted three extensive semi-structured interviews with security domain experts from December 2013 to March 2014. The interviewees were recruited from the departments of research, penetration testing, and risk management consulting of SBA Research[7], a well-established security competence center in Vienna, Austria. All of them have earned degrees in computer science, hold information security certificates such as CISSP, CSSLP, CISA, AMBCI, and CEH, and have several years of working experience in industry projects in their respective fields.

Each interview lasted between 2.5 and 3 hours and comprised 18 questions that were organized into four thematic areas, namely, knowledge base (3 questions), simulation (3 questions), application example (5 questions), and optimization and decision support (7 questions), plus 6 questions concerning background and job context. Before each interview section, a scripted presentation explained the respective part of the methodology and illustrated the practical applicability by means of examples. This was followed by the interview questions and an open

[7] http://sba-research.org

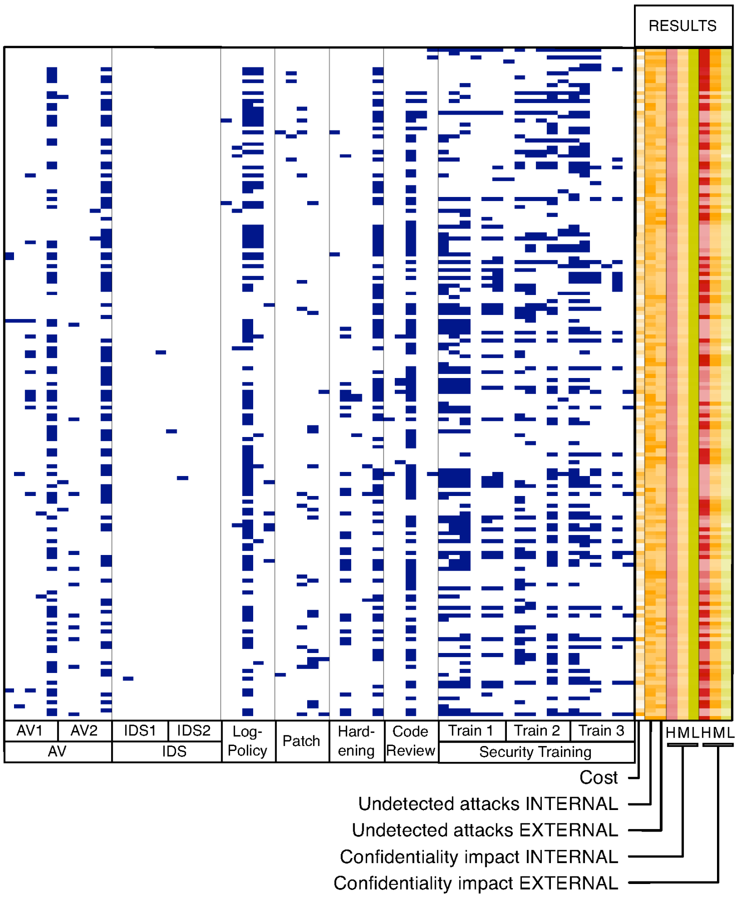


Fig. 11 Heatmap for multiple attackers in Scenario 2

discussion. The interviews were conducted in German. In the following, responses and feedback from the interviewees will be discussed in three groups referring to their overall assessment of our methodology, modeling issues, and decision support aspects.

7.1 Overall assessment

Although numerous risk management methods have been developed in the information security literature, little evidence regarding the usage or the effectiveness of these methods is available (Papadaki and Polemi, 2007). In practice, organizations often create their own methods or adapt existing information security risk management practices to their business environment and culture (Papadaki and Polemi, 2007). Hence, there is no established information security risk management "benchmark" that could be used as a reference to compare our approach against (for a more general comparison of related approaches and their scope, see Sect. 2).

The interviewees' assessments of our methodology, our general approach, and the prototype of the decision support system were positive. One of the interviewees, for instance, concluded that the capability of modeling individual configurations and assessing the security through simulation is in itself highly valuable for operational security personnel. All interviewees stated that integrating this simulation capability into a larger optimization and decision-support framework is a promising approach. Furthermore, they agreed that optimizing a complete IT infrastructure, rather than deciding on individual security measures, is reasonable, given that the weakest element in a security architecture determines the overall security of the system. Interestingly, an interviewee who works as an industry consultant was highly vocal about this viewpoint, whereas the penetration tester stipulated that such a "non-reductionist" approach was valid, but not necessarily essential. This may be explained by the differences in the scope of their respective perspectives (i.e., security management vs. analysis of particular weaknesses). All interviewees highlighted the possibility to model the system, analyze effective portfolios, and select suitable controls in a single integrated environment.

Based on their (extensive) practical experience, all interviewees shared the view that it is imperative to ground security decisions in a threat model that reflects attackers' heterogeneous motivations, resources, capabilities, and points of access. They considered the attacker behaviors produced by our simulation model to be realistic. One of the interviewees underlined the practical relevance of skilled internal attackers (which we included in our application example in Sect. 6.2).

Overall, the interviewees found that the methodology provides valuable insights that could be beneficial in the context of their work. They emphasized the importance of presenting the results in a format that is suitable for the target audience (i.e., both security analysts and managers). One of the interviewees shared anecdotal evidence that managers are usually only cursorily interested in a few key indicators, whereas security analysts develop and base their recommendations on detailed technical considerations. This highlights the value of our methodology, as it aims to unite both perspectives in a decision-support tool that presents high-level decision criteria to management, but also links the high-level criteria optimization to detailed technical information generated in the course of the simulation.

An interviewee working in the security consulting domain raised the point that for some types of controls, it is best practice to deploy them (e.g., anti-virus software) and, thus, the question of whether or not to deploy a control does not necessarily arise. However, he further explicated that in this case, the value of the optimization lies in its ability to compare the relative overall effectiveness of various alternatives for these "best practice controls". Note that existing controls or controls for which an investment decision has already been made can be included in our system model as fixed elements in a straightforward manner. Deployment of best practice controls could also be enforced through restrictions on permissible genotypes in future versions of our implementation. The same holds for compliance with best practices that could be easily incorporated as an additional optimization objective.

Finally, the interviewees expressed their view that our approach is not limited to IT security, and in their opinion, it could also be applied in different domains such as the security area in general, quality management, industrial production, or availability analysis.

7.2 Modeling issues

The interviewees perceived the model structure of the attack and security KBs as generic and expressive, and hence, considered them applicable for a wide range of security domains. However, the considerable effort required to create and maintain these models was frequently raised as an important practical issue. Building such a comprehensive security KB is time-consuming and requires considerable expert knowledge. In this context, it was acknowledged that referring to existing sources, such as the well-established list of CAPEC attack patterns used in our application example, make the efficient formal definition of attack actions easier. Moreover, once defined, the security knowledge can be reused and shared among multiple organizations. The creation and maintenance of the security KB can also, for instance, be organized as a community effort or could be delivered as a commercial service. By leveraging a centralized security knowledge repository, organizations would only need to model and maintain the organization-specific system knowledge. This process – i.e., modeling an IT infrastructure and subsequently updating the model continuously to reflect changes made – could be partly automated or linked to existing internal repositories.

Other discussions revolved around the completeness and correctness of the modeled knowledge. To this end, it is necessary to ensure that (i) the system model reflects the current configuration of the real-world system, and (ii) the security knowledge includes a comprehensive and correct set of relevant actions and controls. The first issue must be addressed by the modeler for each particular application. This can be supported through automated tools.

The second issue is arguably more difficult. The interviewees stressed the importance of reconnaissance and discovery phases in most attacks. As illustrated by the *Scan Network* action in the example application, our framework allows for such attack actions. Discovery actions can be successful or fail, and the attacker will only obtain the necessary knowledge to commence with an attack upon success. Accounting for unknown attack actions, however, is – by their very definition – challenging. As demonstrated by the *Zero Day* action in our example, unknown attacks can be modeled generically if their basic characteristics are conceivable. For such types of unknown attack vectors, the particular weaknesses that are exploited and the detailed attack mechanism are not known in advance, but their preconditions and likely effect (postconditions) are known. Entirely unknown attack actions, i.e., completely new ways of attacking a system, cannot be modeled in advance, but it would be possible to introduce randomly generated attack actions to perform hypothetical robustness assessments. This rather speculative approach would, however, greatly raise the complexity of the optimization problems.

Furthermore, an interviewee reckoned that obtaining exact values for the *likelihood of success* of an action as well as for the *controls' effectiveness* can be difficult in practice. This is a valid point in principle, but in many cases, necessary data can be estimated with sufficient accuracy. For instance, the likelihood that a brute force password attack (i.e., enumerating all combinations) succeeds (within a given timeframe) or the effectiveness of a password policy (that prescribes a certain minimum character length, alphabet size, etc.) can be determined mathematically. Still, we agree that precise numbers are often unknown and approximate value estimates are, therefore, necessary. For various control types, such as firewalls, anti-virus software, and intrusion detection systems, the modeler can derive such effectiveness values from existing benchmarks, statistics, and empirical evidence on the prevalence of successful exploits. For the remaining actions and controls, reasonable assumptions must be made. However, we found that optimization results are usually fairly robust against small deviations in the likelihood of success and control effectiveness.

7.3 Decision support aspects

Interviewees deemed the heatmaps, radar charts, and interactive parallel coordinate visualizations helpful for visual comparisons of particular portfolios and trading off their benefits and drawbacks. Still, they recommended several visual modifications. Most of these suggestions are already incorporated in the prototype version presented in this paper (see Sect. 4.5). The development of a comprehensive management dashboard suggested by an interviewee remains an open issue for future development.

The interviewees also expressed their opinion that analyzing specific scenarios – such as attacks on a particular asset by a particular attacker – is helpful, but may not be sufficient for making security decisions. One interviewee emphasized that it is necessary to conduct both analyses of scenarios with multiple attacker profiles (see the example in Sect. 6.3) and in-depth analyses for particularly critical assets and attacker profiles (see the example in Sect. 6.2). This adds complexity to the process as decision makers ultimately must deal with multiple sets of proposed efficient security portfolios obtained for the various scenarios. Although this process is more laborious and time-consuming than selecting from a single set of proposed efficient solutions, the interviewee concluded that such a process fosters learning, insights, and an understanding of the interactions between security controls in various contexts and, thus, contributes to a more profound security control portfolio investment decision.

8 Conclusions

This paper has introduced a comprehensive methodology for model-driven information security optimization that aims at facilitating informed choices about effective and efficient means to secure an organization's IT infrastructure. Our approach relies on formal modeling of security knowledge, explicit modeling of IT systems and threats, discrete-event simulation of attacks, and a multi-objective genetic algorithm that identifies Pareto-efficient security control portfolios. The methodology is implemented in a decision support system that provides decision makers with three visualizations for the interactive exploration of the solution space. The applicability of the approach is demonstrated through two sample scenarios.

To validate our approach, we conducted extensive semi-structured interviews with experts from multiple security domains. Their overall assessment was positive, with particular acknowledgment of the knowledge base, the attacker-centric approach, and the potential of our decision support system to act as a valuable communication tool for different groups of stakeholders. Interviewees also pointed out that the methodology requires decision makers to enumerate and valuate the organization's assets and to explicate assumed threats which should raise situational awareness in terms of the current level of security, relevant threats, and critical assets that need to be protected. Furthermore, they highlighted that our approach can provide a better understanding of possible routes of attack and improve the documentation of and the justification for IT security investments and, finally, it also improves organizational alignment between management and IT operations by providing a communication tool that both IT professionals and management can relate to.

An obvious practical implication of our research is that such an approach necessitates a more collaborative model for security decision making. Its implementation therefore requires a cultural shift toward a more transparent and consensual approach grounded in a shared understanding of security issues and threats which ultimately fosters stronger management involvement in security decision making. However, these long-term implications will only materialize if the stakeholders gain trust in model-based security decision support. To this end, several limitations of our current research still need to be addressed. First, the potential of our methodology in a real-world setting hinges upon a comprehensive security KB. Building such a KB requires considerable expert knowledge, is time-consuming, and hence expensive for any individual organization. However, a community-based effort to create and maintain a shared repository for security knowledge could overcome this limitation. Furthermore, the simulation approach also requires a detailed and up-to-date model of the system to be secured. While creating and maintaining such a model manually is nearly impossible in large organizations, this process could be efficiently supported through automated tools and integration with existing internal repositories. A final limitation of our approach lies in its reliance on an accurate model of attacker behavior. In order to create and validate these models, a deeper understanding of behavioral aspects of attacks is required which necessitates interdisciplinary empirical research that takes into account technological, psychological, sociological, and economic perspectives.

Additional directions for future research were derived from the expert interviews we conducted. Compliance coverage, which can be introduced either as an optimization criterion or as constraints on feasible genotypes, constitutes one such possible direction. Integrating business continuity into the framework by simulating the impact of attacks on business processes would be another interesting extension. Finally, the approach could be transferred to other domains with similar security requirements, such as critical infrastructures and production security.

Acknowledgements The work presented in this paper has been developed within the project MOSES[3], which was funded by the Austrian Science Fund (FWF) under grant P23122-N23. The research was carried out at Secure Business Austria, a COMET K1 program competence center supported by the Austrian Research Promotion Agency (FFG). Computational results have been achieved using the Vienna Scientific Cluster (VSC).

References

Ammann P, Wijesekera D, Kaushik S (2002) Scalable, graph-based network vulnerability analysis. In: Proceedings of the Conference on Computer and Communications Security, ACM, pp 217–224

Baker WH, Wallace L (2007) Is information security under control? Investigating quality in information security management. IEEE Security & Privacy 5(1):36–44

Barlette Y, Fomin VV (2010) The adoption of information security management standards. In: Information Resources Management: Concepts, Methodologies, Tools and Applications, IGI Global, pp 69–90

Bistarelli S, Fioravanti F, Peretti P (2006) Defense trees for economic evaluation of security investments. In: Proceedings of the International Conference on Availability, Reliability and Security, IEEE Computer Society, pp 416–423

BSI (2013) BSI-Standards. Tech. Rep., German Federal Office for Information Security

Chi SD, Park JS, Jung KC, Lee JS (2001) Network security modeling and cyber attack simulation methodology. In: Varadharajan V, Mu Y (eds) Information Security and Practice (LNCS 2119), Springer, pp 320–333

Cohen F (1999) Simulating cyber attacks, defences, and consequences. Computers & Security 18(6):479–518

Cook D, Hofman H, Lee EK, Yang H, Nikolau B, Wurtele E (2007) Exploring gene expression data, using plots. Journal of Data Science 5(2):151–182

Dahl OM, Wolthusen SD (2006) Modeling and execution of complex attack scenarios using interval timed colored petri nets. In: Proceedings of the International Workshop on Information Assurance, IEEE, pp 157–168

Dalton GC, Mills RF, Colombi JM, Raines RA (2006) Analyzing attack trees using generalized stochastic Petri nets. In: Proceedings of the Information Assurance Workshop, IEEE, pp 116–123

Deb K, Pratap A, Agarwal S, Meyarivan T (2000) A fast elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2):182–197

Draper MD, Livnat Y, Riesenfeld RF (2009) A survey of radial methods for information visualization. IEEE Transactions on Visualization and Computer Graphics 15(5):759–776

Economist (2014) Defending the digital frontier: A special report on cyber-security. The Economist, July 12, 2014

Edge KS, Dalton GC, Raines RA, Mills RF (2006) Using attack and protection trees to analyze threats and defenses to homeland security. In: Proceedings of the Military Communications Conference, IEEE, pp 1–7

Ekelhart A, Kiesling E, Grill B, Strauss C, Stummer C (2015) Integrating attacker behavior in IT security analysis: A discrete-event simulation approach. Information Technology and Management 16(3):221–233

Fenz S, Ekelhart A (2011) Verification, validation, and evaluation in information security risk management. IEEE Security & Privacy Magazine 9(2):58–65

Fenz S, Ekelhart A, Neubauer T (2011) Information security risk management: In which security solutions is it worth investing? Communications of the Association for Information Systems 28:329–356

Franqueira VNL, Lopes RHC, van Eck P (2009) Multi-step attack modelling and simulation (MsAMS) framework based on mobile ambients. In: Proceedings of the Symposium on Applied Computing, ACM, pp 66–73

Gettinger J, Kiesling E, Stummer C, Vetschera R (2013) A comparison of representations for discrete multi-criteria decision problems. Decision Support Systems 54(2):976–985

Gordon LA, Loeb MP (2002) The economics of information security investment. ACM Transactions on Information and System Security 5(4):438–457

Gupta M, Rees J, Chaturvedi A, Chi J (2006) Matching information security vulnerabilities to organizational security profiles: A genetic algorithm approach. Decision Support Systems 41(3):592–603

Hoo S (2000) How Much is Enough: A Risk Management Approach to Computer Security. PhD Thesis, Consortium for Research on Information Security and Policy (CRISP), Stanford University

Inselberg A (2009) Parallel Coordinates: Visual Multidimensional Geometry and its Applications. Springer

Islam T, Wang L (2008) A heuristic approach to minimum-cost network hardening using attack graph. In: Proceedings of the Conference on New Technologies, Mobility and Security, IEEE, pp 1–5

ISO (2013) ISO/IEC 27001:2013: Information technology, security techniques, information management systems, requirements. Tech. Rep., International Organization for Standardization/International Electrotechnical Commission

Jaisingh J, Rees J (2001) Value at risk: A methodology for information security risk assessment. In: Proceedings of the Conference on Information Systems and Technology, INFORMS, pp 3–4

Kaspersky (2014) IT security risks survey 2014: A business approach to managing data security threats. http://media.kaspersky.com/en/IT_Security_Risks_Survey_2014_Global_report.pdf (Accessed July 11, 2015)

Keeney RL (2013) Identifying, prioritizing, and using multiple objectives. European Journal on Decision Processes 1(1-2):45–67

Kiesling E, Ekelhart A, Grill B, Strauss C, Stummer C (2013a) Simulation-based optimization of information security controls: An adversary-centric approach. In: Pasupathy R, Kim SH, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the Winter Simulation Conference, IEEE Computer Society, pp 2054–2065

Kiesling E, Ekelhart A, Grill B, Strauss C, Stummer C (2013b) Simulation based optimization of IT security controls: Initial experiences with metaheuristic solution procedures. In: Fink A, Geiger M (eds) Proceedings of the Workshop of the EURO Working Group on Metaheuristics, pp 18–20

Kiesling E, Ekelhart A, Grill B, Stummer C, Strauss C (2014) Evolving secure information systems through attack simulation. In: Proceedings of the Hawaii International Conference on System Science, IEEE Computer Society, pp 4868–4877

Kiesling E, Ekelhart A, Grill B, Stummer C, Strauss C (2015) Multi-objective evolutionary optimization of computation-intensive simulations: The case of security control selection. In: Proceedings of the 11th Metaheuristics International Conference, pp 1–3

Lotov A, Miettinen K (2008) Visualizing the Pareto frontier. In: Branke J, Deb K, Miettinen K, Slowinski R (eds) Multiobjective Optimization (LNCS 5252), Springer, pp 213–243

Lukasiewycz M, Glaß M, Reimann F, Teich J (2011) Opt4J: A modular framework for meta-heuristic optimization. In: Proceedings of the Conference on Genetic and Evolutionary Computation, ACM, pp 1723–1730

Luke S, Cioffi-Revilla C, Panait L, Sullivan K, Balan G (2005) MASON: A multiagent simulation environment. Simulation 81(7):517–527

Ma Z, Smith P (2013) Determining risks from advanced multi-step attacks to critical information infrastructures. In: Luiijf E, Hartel P (eds) Critical Information Infrastructures Security (LNCS 8328), Springer, pp 142–154

Mauw S, Oostdijk M (2006) Foundations of attack trees. In: Won D, Kim S (eds) Information Security and Cryptology (LNCS 3935), Springer, pp 186–198

McAfee (2014) Net losses: Estimating the global cost of cybercrime 2014. `http://www.mcafee.com/de/resources/reports/rp-economic-impact-cybercrime2.pdf` (Accessed July 11, 2015)

Mizzi A (2005) Return on information security investment. Are you spending enough? Are you spending too much? `http://security.ittoolbox.com/documents/return-on-information-security-investment-14513` (Accessed July 11, 2015)

Moore A (2001) Attack modeling for information security and survivability. Tech. Rep., Software Engineering Institute, Carnegie Mellon University

National Bureau of Standards (1979) Guideline for automatic data processing risk analysis. Tech. Rep., Institute for Computer Science and Technology, National Bureau of Standards

NIST (2011) Managing information security risk: Organization, mission, and information system view. Tech. Rep., NIST SP 800-39, National Institute of Standards and Technology, U.S. Department of Commerce

Neubauer S, Stummer C, Weippl E (2006) Workshop-based multiobjective security safeguard selection. In: Proceedings of the International Conference on Availability, Reliability and Security, IEEE Computer Society, pp 366–373

Ou X, Boyer WF, McQueen MA (2006) A scalable approach to attack graph generation. In: Proceedings of the Conference on Computer and Communications Security, ACM, pp 336–345

Panchenko A, Pimenidis L (2006) Towards practical attacker classification for risk analysis in anonymous communication. In: Leitold H, Markatos EP (eds) Communications and Multimedia Security (LNCS 4237), Springer, pp 240–251

Papadaki K, Polemi N (2007) Towards a systematic approach for improving information security risk management methods. In: Proceedings of the International Symposium on Personal, Indoor and Mobile Radio Communications, IEEE, pp 1–4

Pieters W (2011) Representing humans in system security models: An actor-network approach. In: Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 2(1):75–92

Ritchey RW, Ammann P (2000) Using model checking to analyze network vulnerabilities. In: Proceedings of the IEEE Symposium on Security and Privacy, IEEE, pp 156–165

Sawilla RE, Ou X (2008) Identifying critical attack assets in dependency attack graphs. In: Jojadia S, Lopez J (eds) Computer Security (LNCS 5283), Springer, pp 18–34

Schneier B (2000) Secrets & Lies: Digital Security in a Networked World, Wiley

Stoneburner G, Goguen AY, Feringa A (2002) Risk management guide for information technology systems: Recommendations of the National Institute of Standards and Technology. Tech. Rep., NIST SP 800-30, National Institute of Standards and Technology, U.S. Department of Commerce

Strauss C, Stummer C (2002) Multiobjective decision support in IT-risk management. International Journal of Information Technology and Decision Making 1(2):251–268

Stummer C, Kiesling E, Gutjahr WJ (2009) A multicriteria decision support system for competence-driven project portfolio selection. International Journal of Information Technology and Decision Making 8(2):379–401

Tunçalp D (2014) Diffusion and adoption of information security management standards across countries and industries. Journal of Global Information Technology Management 17(4):221–227

Vetschera R (2013) Negotiation processes: An integrated perspective. European Journal on Decision Processes 1(1-2):135–164

Vincke P (1992) Multicriteria Decision-aid, Wiley

Wang J, Chaudhury A, Rao HR (2008) Research note: A value-at-risk approach to information security investment. Information Systems Research 19(1):106–120

Wang L, Noel S, Jajodia S (2006) Minimum-cost network hardening using attack graphs. Computer Communications 29(18):3812–3824

Wielemaker J, Schrijvers T, Triska M, Lager T (2012) SWI-Prolog. Theory and Practice of Logic Programming 12(1-2):67–96

Zitzler E, Laumanns M, Thiele L (2002) SPEA2: Improving the Strength Pareto Evolutionary Algorithm for multiobjective optimization. In: Giannakoglou K, Tsahalis D, Periaux J, Papailiou K, Fogarty T (eds) Evolutionary Methods for Design, Optimisation and Control, CIMNE, pp 1–6