Risk-Aware Decision Support with Constrained Goal Models

Nikolaos Argyropoulos, Konstantinos Angelopoulos,
Haralambos Mouratidis, and Andrew Fish

Centre for Secure, Intelligent and Usable Systems,
University of Brighton,
School of Computing, Engineering and Mathematics,
Brighton, UK
{n.argyropoulos, k.angelopoulos, h.mouratidis, andrew.fish}@brighton.ac.uk

Abstract.
Purpose - The selection of security configurations for complex information systems is a cumbersome process. Decision-making regarding the choice of security countermeasures has to take into consideration a multitude of, often conflicting, functional and non-functional system goals. Therefore, a structured method to support crucial security decisions during a system's design that can take account of risk whilst providing feedback on the optimal decisions within specific scenarios would be valuable.
Approach - Secure Tropos is a well established security requirements engineering methodology, but it has no concepts of Risk, whilst Constrained Goal Models are an existing method to support relevant automated reasoning tasks. Hence we bridge these methods, by (i) extending Secure Tropos to incorporate the concept of Risk, so that the elicitation and analysis of security requirements can be complimented by a systematic risk assessment process during a system's design time; (ii) supporting the reasoning regarding the selection of optimal security configurations with respect to multiple system objectives and constraints, via Constrained Goal Models.
Findings - As a means of conceptual evaluation, to give an idea of the applicability of the approach and to check if alterations may be desirable, a case study of its application to an e-government information system is presented. The proposed approach is able to generate security mechanism configurations for multiple optimisation scenarios that are provided, whilst there are limitations in terms of a natural trade-off of information levels of risk assessment that are required to be elicited.
Originality - The proposed approach adds additional value via its flexibility in permitting the consideration of different optimisation scenarios by prioritising different system goals and the automated reasoning support.

Keywords: Information Security, Security Requirements, Decision Making, Constraint Goal Models, Secure Tropos

1 Introduction

The advances of technology have dramatically increased user expectations of modern information systems. The continuous growth of the number of goals that these systems are expected to satisfy, as well as the complexity of their architectures, render software configuration (or reconfiguration) a challenging process. In particular, information systems which are exposed to cyber-threats must be able to respond to continuous changes in their environment that could put their valuable assets at risk.

The selection of appropriate security configurations should take into consideration the threat landscape in which the system will operate. Therefore, the effects of vulnerabilities and threats towards a system's goals, and their mitigation by security countermeasures, should play an important role during the system's design process (Viduto et al., 2012). The ever-changing nature of the threat landscape is amplified by new paradigms in information system architecture (e.g., cloud computing, Internet of Things) (Islam et al., 2017). In such volatile environments the risk posed by a threat can vary greatly, depending on the impacted system component or the likelihood of a vulnerability being exploited. Therefore, a flexible approach towards risk-aware decision-making is crucial during system design, especially with regards to the system's security countermeasure configuration. An attempt to provide risk-aware decision support should also be able to take into account trade-offs between security and other functional and non-functional system goals. Thus, perhaps unsurprisingly, striking a balance between effective risk management and functional system design is a challenging endeavour.

To make progress in addressesing these challenges, in this paper we present an extension of Secure Tropos and propose a methodology to support risk-aware decision-making for the design of secure system configurations. Secure Tropos (Mouratidis and Giorgini, 2007) is a security-oriented extension of the Tropos methodology (Bresciani et al., 2004), which makes use of goal models to support the elicitation and analysis of security requirements from the early stages of the system development life-cycle. Other than its support for security modelling in an explicit and structured manner, Secure Tropos has been selected as the basis of our approach due to: (i) its social concepts (e.g., actors, goals, dependencies) and analysis capabilities during the early requirements stage; (ii) the simultaneous consideration of security along with the other requirements of the system-to-be; and (iii) its ability to support the design stage of system development, through the mapping of abstract security constraints and threats to specific security mechanisms (Argyropoulos et al., 2015).

In terms of novelty, firstly risk related concepts and attributes are integrated into Secure Tropos, thereby allowing designers to express the level of security of their systems as cost-functions. Next, we propose the use of the Analytic Hierarchy Process (AHP) (Saaty, 1988) to estimate the likelihood of threats to be manifested. Our approach provides a new framework that selects optimal security configurations with respect to the severity of threats and the priorities of other goals. More specifically, we express the level of mitigation of each threat and

other goals of the systems (e.g. cost and performance) as cost-functions that our proposed framework optimises according to a prioritised order of cost function; that is, the best adaptations are selected according to the most important (most highly prioritised) cost function and the best among those are selected according to the second-most important cost function, etc. This paper builds on the work of (Argyropoulos, Angelopoulos, Mouratidis and Fish, 2017) by improving the formalisation and description of the risk calculation formulas, enhancing the discussion of the proposed approach (e.g. of AHP, semi-automation), as well as applying it to scenarios that express "real-life needs" of system stakeholders.

In terms of paper organisation, in Section 2 we first introduce central concepts of all of the relevant research areas. Then, in Section 3 we describe: (i) how the Secure Tropos meta-model is extended in order to support the notion of risk; (ii) the instantiation of the basic risk assessment variables. In Section 4 we illustrate our approach through a case study, providing a means of conceptual evaluation aimed at eliciting possible limitations of future requirements for alterations of the methodology, whilst providing some belief in the method's practical applicability. To aid the flow of the paper, we leave the discussion of related works until Section 5. Lastly, in Section 6, we present discussions, conclusions and directions of future work.

2 Research Baseline

In this section, we provide a research baseline for the Secure Tropos approach, constrained goal models, and risk management, which are the main building blocks of our proposed approach.

2.1 Goal Models

Secure Tropos, and therefore our approach, adopts the principles of Goal Oriented Requirements Engineering (GORE). The centrepiece of GORE is the concept of goal (Dardenne et al., 1993) that captures the intentions of stakeholders. The initial goals are generally gradually refined into more detailed goals by making use of the facility to express AND/OR boolean relationships. The refinement process ends when each goal is refined into detailed tasks that can be assigned to a human or software component. These tasks are called plans in the Secure Tropos terminology. Since goals and plans can only express functional requirements of the system, the concept of softgoal (Chung et al., 2000) is additionally used in order to express non-functional requirements. In Giorgini et al. (2003), the fulfilment of a goal (or its lack of fulfilment) is characterised by the use of the four propositions: full satisfaction (FS), partial satisfaction (PS), full denial (FD) and partial denial (PD). In this work, in order to simplify the setup and to focus on the core ideas, we do not consider partial propositions; they could be an potential extension after the benefits of the new method is established.

2.2 Security Requirements Engineering

Secure Tropos (Mouratidis and Giorgini, 2007) is a goal-oriented security requirements engineering approach and, as such, it is able to support the elicitation and analysis of security requirements from the early stages of the system development life-cycle. It utilises standard goal-oriented requirements engineering concepts (e.g., actors, goals, dependencies) but it also introduces concepts from the domain of security engineering (e.g., security constraints, threats, security mechanisms). Secure Tropos facilitates system design and (security) requirements elicitation through a number of interrelated modelling views. The *Security Requirements* view is used to present the goal decomposition of each system actor and the dependencies between them. Additionally, security constraints and threats are identified and connected to goals and resources within the same modelling view; potential security mechanisms that can satisfy the identified constraints and threats are also identified. The *Security Attacks* view consists of an additional diagram for each threat that is identified within the Security Requirements view; this decomposes each threat in order to identify its attack methods, the system vulnerabilities that they exploit, as well as the coverage provided by the proposed security mechanisms against such vulnerabilities. The use of online threat repositories (e.g., CAPEC (MITRE, 2017)) and the consultation of security experts are recommended for the identification of threats, attack methods and vulnerabilities and the derivation of sets of potential security mechanisms. A detailed presentation of the components and modelling views of the Secure Tropos methodology can be found in Mouratidis et al. (2016).

2.3 Constrained Goal Models

Goal models often present high variability, expressed by multiple alternative solutions to fulfil one or more goals. One of the tasks of GORE is to decide which of these alternatives should be implemented, and which should not be, in the system-to-be. Given the nature of goal models, each goal represents a predicate that can be related to other predicates via AND/OR relationships. Therefore such relationships between goals can naturally be used to construct first order logic formulas.

In order to elaborate on complex aspects of system designs, captured by goal models, additional attributes can be assigned to different components of the models. In this work we introduce a number of attributes to try to quantitatively capture certain aspects of risk, security coverage and non-functional goals. Thus, each alternative solution, in terms of choices of security mechanisms, leads to a goal model with different values for each of the variables captured by the newly-introduced attributes. Hence, goal reasoning in our approach corresponds to finding a solution to a maximum satisfiability (MAX-SAT) problem.

To solve such problems we must turn our attention to the field of satisfiability and optimisation modulo theories (SMT/OMT). There, the combination of the different variables are captured by formulas associated with linear equations that must be optimised by any solution found for the satisfiability problem. The integration of SMT/OMT with goal models has been implemented by

Constrained Goal Models (CGMs) (Nguyen et al., 2016). Such goal models permit the definition of: a) multiple variables associated with the modelled goals; and b) linear equations composed of these variables that should be optimised. Therefore, alongside the satisfiability problem, which is native to goal models, a multi-objective optimisation problem should be solved in parallel. This is performed via the use of a scalable external reasoner, OptiMathSAT (Sebastiani and Trentin, 2015), which is invoked to find optimal solutions over CGMs.

The use of such a reasoner enables flexibility in the optimisation process so that system designers and stakeholders can decide: (i) which variables capture critical aspects of the system and should, therefore, be included in the formulas; and (ii) the priority of each of the selected variables in the optimisation process. As a result, the application of the reasoner can produce a number of system configurations depending on the selected variables and their prioritisation. This enables the construction of a number of scenarios during the decision support step of the approach, each of which produces a different system configuration in terms of the selected security mechanisms. Each of the resulting configurations can then be used to produce a different business process instance by following the rest of the framework's steps.

2.4 Risk Management

In the field of information security, a risk expresses the potential of a threat to exploit vulnerabilities of organisational assets and as a result harm the organisation (ISO/IEC, 2008). Risk management is a set of coordinated activities performed by an organisation to minimise the effects of risks (ISO/IEC, 2014).

Risk assessment is the initial phase of the risk management process, during which organisations elicit potential threats and the vulnerabilities they exploit to threaten the functionality of their systems. The risk introduced by such vulnerabilities is evaluated and security countermeasures for reducing or eliminating the identified risk are recommended (Stoneburner et al., 2002). Values for the impact and the likelihood of each identified vulnerability can be estimated using either quantitative or qualitative metrics (Blakley et al., 2001). The consensus approach for assigning a value to the risk introduced by each vulnerability is by calculating the product of its impact and likelihood (Open Web Application Security Project, 2015; Stoneburner et al., 2002). The overall risk introduced by a threat can then be calculated as the sum of the individual risk values of each of its associated vulnerabilities.

Risk reduction, via the use of countermeasures, is amongst the most established strategies for risk mitigation. Countermeasures need to be prioritised in terms of the coverage that they provide against each risk, but also in terms of their contribution towards other non-functional objectives of the system (e.g., financial cost, technical constraints, usability) (ISO/IEC, 2008). The risk remaining after the application of a risk mitigating strategy is known as the residual risk. The final phase of the risk management process involves the continuous evaluation and assessment of the implemented system throughout its life-cycle,

in order to account for potential changes to its composition and its execution environment.

3 Capturing Risk with Secure Tropos

Secure Tropos introduces a conceptual basis which facilitates security trade-off modelling and analysis (Elahi and Yu, 2007). However, an inherent limitation of all Tropos-based approaches is their lack of precise semantics for the quantitative evaluation of system behaviours, including security and risk coverage (Cailliau and Van Lamsweerde, 2012). Additionally, concepts necessary for the risk analysis process (e.g., risk) are missing. Whilst attempts to align Tropos with risk-related concepts have been developed (Matulevičius et al., 2008), they lack the ability to quantitatively perform risk assessment and support a fine-grained security trade-off analysis. To that end, we extend Secure Tropos with a number of concepts and attributes, as presented in Fig. 1 in a bold, italic font.

3.1 Conceptual Model for Risk

The concept of *Risk* is introduced into the existing Secure Tropos metamodel and connected to the concept of *Threat*, since any threat introduces a certain amount of risk through its associated *Vulnerabilities*. Each vulnerability represents a potential weakness that can be exploited by a threat and can thereby compromise the system's security.

The impact of each vulnerability is captured by the attribute *Impact*, which can be evaluated using a number of different techniques. A common approach used for estimating the impact of vulnerabilities is by using CVSS (Common Vulnerabilities Scoring System) (Mell et al., 2007) and/or historical data. A semi-quantitative scale is often used for the value assignment of a vulnerabilities' impact using discrete values (e.g., 10, 50, 100 to represent low, medium, or high impact) (Viduto et al., 2012). However, in this work we choose to estimate the impact of a particular vulnerability as being the relative impact of the vulnerability with respect to the impact of all of the other vulnerabilities of the system. This means that the higher the value of the impact the more important a vulnerability is deemed to be. Therefore, this allows us to estimate the impact of each vulnerability by applying the Analytic Hierarchy process (AHP) (Saaty, 1980, 1988), a prioritisation approach commonly used in software engineering (Karlsson and Ryan, 1997; Vaidya and Kumar, 2006).

The probability of a vulnerability being exploited for the manifestation of a security attack is captured by the *Likelihood* attribute. Similar to the estimation of a vulnerability's impact, the likelihood's value quantifies how much more probable is the exploitation of a vulnerability by a certain threat compared to another one. Therefore, likelihood represents a different prioritisation of vulnerabilities according to their probability of being exploited; it also can be estimated using AHP. In contrast to its impact value, which is unique for

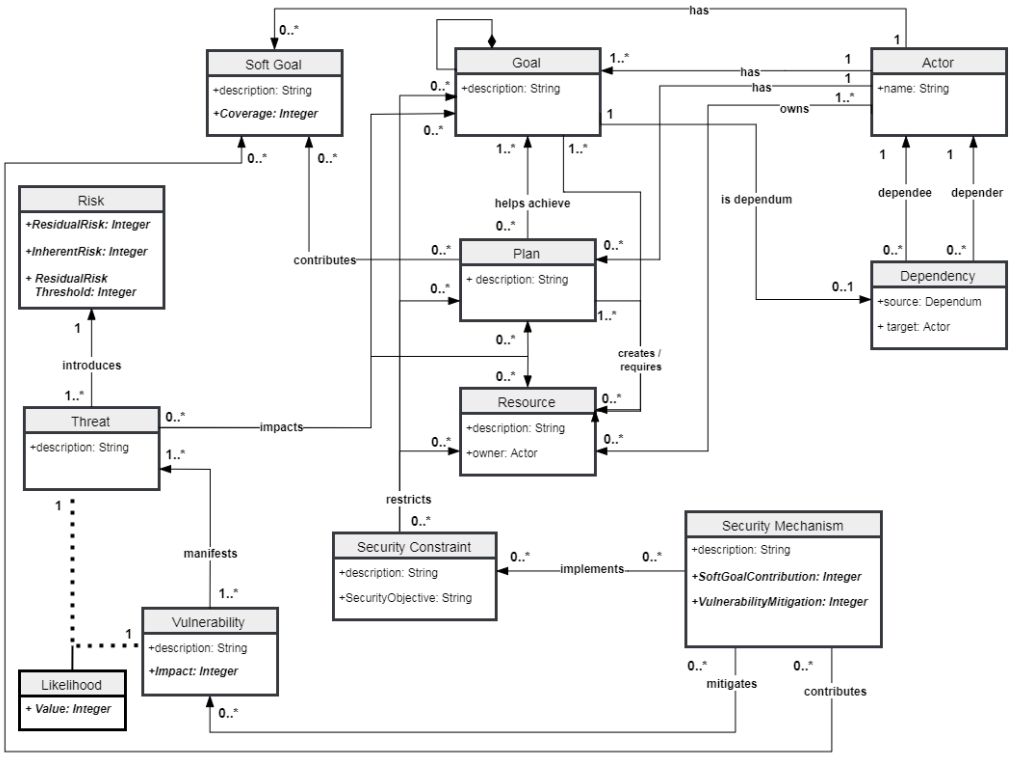


Fig. 1. A partial view of the Secure Tropos metamodel to demonstrate the extensions proposed, including: (i) the addition of new concepts of Risk and Likelihood, with their attributes ResidualRisk, InherentRisk, ResidualRiskThreshold, and Value, respectively; (ii) the extension of existing concepts via the addition of new attributes of Coverage to SoftGoal, Impact to Vulnerability, and SoftGoalContribution and VulnerabilityMitigation to Security Mechanism.

each vulnerability, the likelihood value depends on the combination of a threat-vulnerability pairing, because the same vulnerability can be exploited by more than one threat but with a different likelihood.

The selection of AHP for the assignment of impact and likelihood values allows the ranking of the identified vulnerabilities relative to each other. AHP provides an applicable and intuitive structure to support decision making, making it a popular choice among practitioners(Ishizaka and Labib, 2009). Compared to other approaches, AHP confines the task of value assignment by performing pairwise comparisons between the identified vulnerabilities, instead of defining an arbitrary range of values against which each vulnerability is individually evaluated. However, it also requires a larger overhead in terms of effort and time especially if a large number of vulnerabilities are to be compared. Thus, the

trade-off between the required effort and the precision provided by the application of AHP is an aspect that needs to be considered. Nonetheless, the rest of the introduced steps for the risk assessment process are not dependent on the usage of AHP, as the impact and likelihood values could instead be assigned using any other value assignment technique.

The initial amount of risk introduced by a threat is an aggregation of the risk introduced by each of the vulnerabilities exploited by the threat and is captured by the *InherentRisk* attribute of the *Risk* concept. The amount of risk remaining after risk treatment is applied by the introduction of security mechanisms, is captured by the *ResidualRisk* attribute. Additionally, the attribute *ResidualRiskThreshold* captures, for each threat, the maximum amount of residual risk that would be accepted by the system stakeholders.

The concept of the *Security Mechanism*, which Secure Tropos uses to model technologies utilised to implement the system's security objectives, is extended with a number of attributes. These attributes enable the evaluation of the contribution of each security mechanism towards the achievement of each of the system's soft-goals (*SoftGoalContribution*) and the mitigation of each identified vulnerability (*VulnerabilityMitigation*).

Finally the *Coverage* attribute has been added to the *Soft Goal* concept in order to capture the total coverage provided to each of the system's soft goals by the selected sets of security mechanisms.

3.2 Risk Assessment

Following the development of the extended Secure Tropos meta-model, as previously presented in Fig. 1, we can define functions which will be used to guide the risk-based adaptation process.

Definition 1. *Let $V_1, \ldots, V_n$ denote the vulnerabilities of the system, and let $L_i, I_i \in \mathbb{R}$, with $0 \leq L_i, I_i \leq 1$, denote the Likelihood of V_i being manifested and its Impact, respectively. Let $\overline{V_i} \in \{0, 1\}$ indicate the exploitation of vulnerability V_i by a threat $\overline{V_i} = 1$, or not $\overline{V_i} = 0$.*

*The **Inherent Risk**, R_I, introduced by a threat is defined by:*

$$R_I = \sum_{i=1}^{n} (L_i \times I_i \times \overline{V_i}). \tag{1}$$

Definition 2. *Let $m_i \in \mathbb{N}$ be the number of security mechanisms mitigating vulnerability V_i, and let $M_{ji} \in \mathbb{R}$, with $0 \leq M_{ji} \leq 1$, denote the Vulnerability Mitigation of the j-th security mechanism towards a vulnerability V_i. The **Mitigated Risk** of a threat, R_M, is defined by:*

$$R_M = \sum_{i=1}^{n} \left(L_i \times I_i \times \overline{V_i} \times \sum_{j=1}^{m_i} \frac{M_{ji}}{m_i} \right). \tag{2}$$

The residual risk of each threat is the remainder of its inherent risk when the mitigated risk is subtracted.

Definition 3. *The **Residual Risk** of a threat, R_R is defined as:*

$$R_R = R_I - R_M. \tag{3}$$

It follows that:

Lemma 1.

$$R_R \overset{(3)}{=} R_I - R_M \overset{(1),(2)}{=} \sum_{i=1}^{n} \left[(L_i \times I_i \times \overline{V_i}) \times \left(1 - \sum_{j=1}^{m_i} \frac{M_{ji}}{m_i} \right) \right]. \tag{4}$$

The process for deciding which mechanisms should be implemented has four steps:

Step 1: Security Analysis. The system designers, together with the security engineers, produce Secure Tropos diagrams, as described in Section 2. These models reveal all of the threats to the system's goals and assets under consideration and propose possible alternative solutions, in the form of security mechanisms, to mitigate the threats.

Step 2: Likelihood Estimation. For each vulnerability, a likelihood value is estimated using AHP for each threat. When analysing a vulnerability, the security engineers should assign a likelihood value for each threat that affects this vulnerability.

Step 3: Impact Estimation. For each vulnerability, an impact value is estimated using AHP. To elicit such values, security engineers should perform pairwise comparisons for all of the vulnerabilities and prioritise them based on how much the system will be affected if the considered vulnerability is exploited by a threat.

Step 4. Risk Minimisation. Minimise the Residual Risk by using the optimisation functionality of the extended Secure Tropos. This functionality proposes a set of security mechanisms that minimise the Residual Risk taking also into account other goals, such as Cost and Performance.

Now, new types of attacks are continuously being developed and new vulnerabilities are discovered as software systems evolve. This means that more variables may need to be introduced into our optimisation problem and the previously estimated values for likelihood and vulnerability might need to be updated. Therefore, in this setting the risk management process should be taken to iterative in order to keep the system up-to-date in terms of security decisions throughout its lifecycle. So, in the case of evolving systems the need for re-evaluation could be

considered as a limitation since it adds additional overheads. Nevertheless, the intended scope of the approach proposed in this work was to guide the security choices during the design time of a system to-be. Therefore, while this approach could additionally be applicable to existing/evolving systems throughout their lifecycle, the focus here was restricted to the early design stages of information systems.

4 Case Study

To provide a form of conceptual evaluation of the proposed approach a case study has been developed, focussing on an information system for the registration of citizens to a public swimming pool facility at the Municipality of Athens, Greece. The Secure Tropos framework, the notation of which is presented in Fig. 2, was used to create all relevant system models. A goal model of the system is presented in Fig. 3.

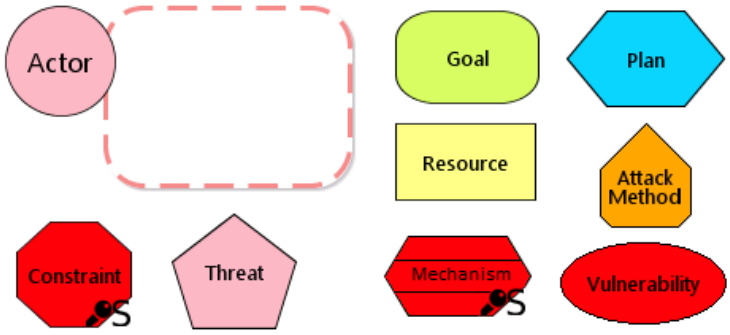


Fig. 2. An overview of the Secure Tropos notation

4.1 System Description

The main participants of the system, declared to be actors at the goal model level, can interact with each other, in the model, via dependency relationships, to achieve their goals. More specifically, the actors considered in this system are: (i) the *Citizen*, aiming to register to be able to use the swimming pool facilities, (ii) the medical *Clinic* that examines the citizen and issues a medical certificate, (iii) the *Municipality of Athens Citizen Services (MACS)* system that citizens can use to request and store certificates, (iv) the *Swimming Pool Information System* that gathers copies of the necessary certificates, registers citizens and tracks their usage of the facilities, and (v) the *Swimming Pool Administrator* that verifies the validity of the citizen's certificates and approves their registration to the facilities. With the collaboration of the system's designers, the goals

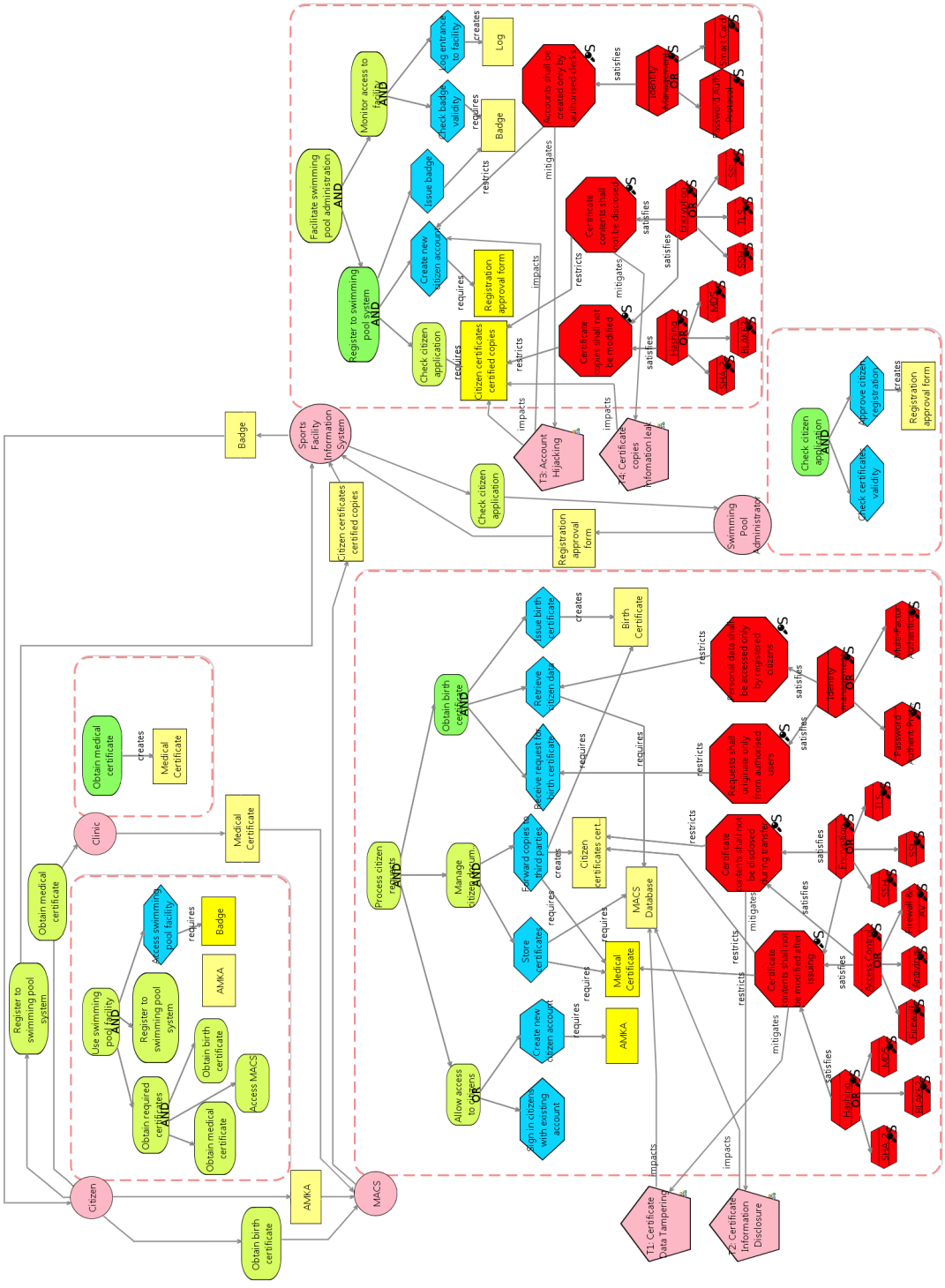


Fig. 3. An extended Secure Tropos goal model of the Swimming Pool Administration information system

of the participating system actors were further decomposed as sub-goals and plans, and the documents and infrastructure created and/or utilised throughout the process were captured as resources. Additionally, non-functional goals (soft-goals) that the overall system should satisfy were defined by its stakeholders. More specifically, the first non-functional goal was to keep the implementation costs at a minimum and the second was to maintain a low system complexity in order not to introduce significant overheads in terms of system performance.

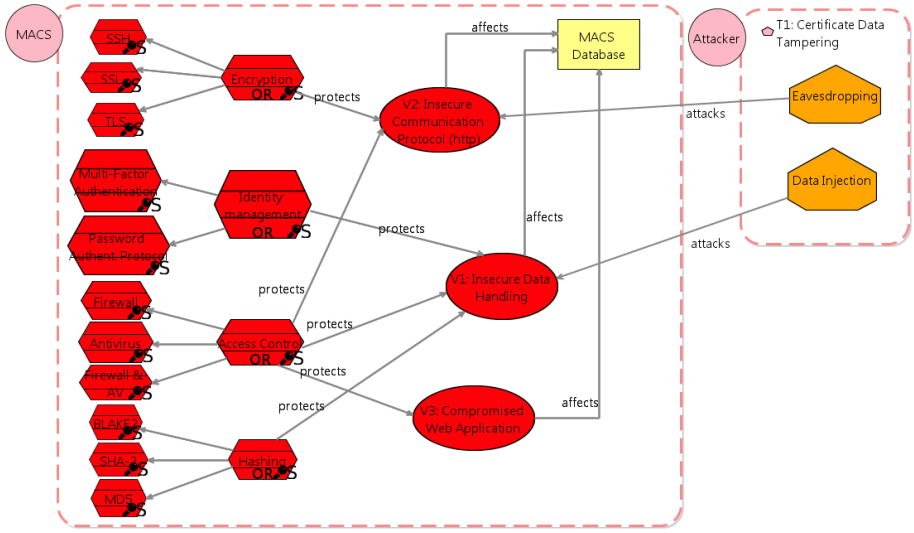
4.2 Application

Following Step 1 of the recommended risk management process, presented in Section 3.2, the security requirements of the system were elicited in the form of security constraints and potential threats, along with the vulnerabilities they exploit. Such security constraints formed the basis upon which the security analysis of the system was performed. The security constraints, restricting certain goals or resources of the system, were identified by the system stakeholders and connected to the relevant model elements during the security analysis process. For instance, *"Certificate contents shall not be modified after issuing"*, was a constraint identified for the EMACS system and this is connected to the resources representing the medical and birth certificates at the goal model level. Each of the identified security constraints were also assigned to the type of security objective (e.g., authentication, authorisation, confidentiality, integrity, availability) that they accomplish.

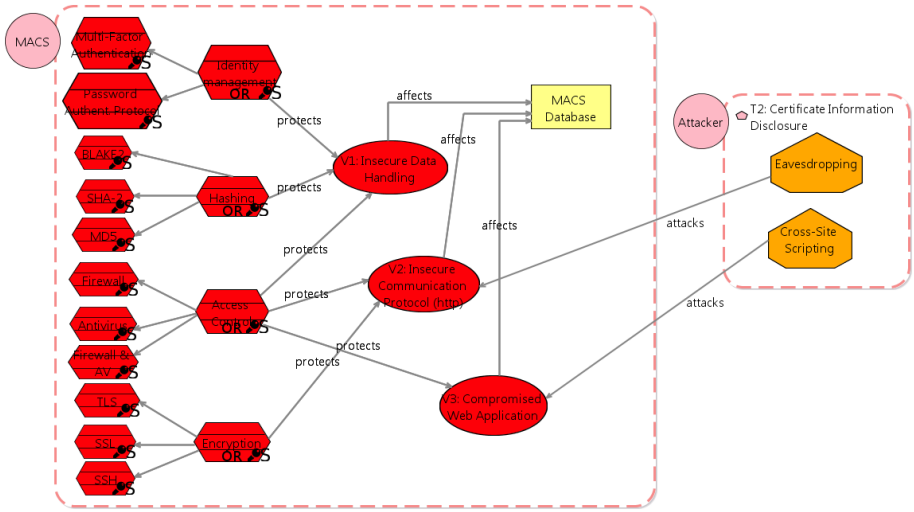
Through the use of relevant resources (e.g. CAPEC (MITRE, 2017)), a number of threats were identified and connected to elements of the system that they can potentially impact. For instance, the threat of *"Account Hijacking"* was identified for the Swimming Pool Information System; this threat can potentially impact the accomplishment of the *"Create citizen account"* plan and the *"Citizen certificates certified copies"* resource. A further breakdown of the threat manifestation and countermeasures for each of the identified threats is provided by the Security Attacks modelling view, as explained below. Finally, a variety of security mechanisms were proposed in order to both satisfy the system's security objectives and mitigate the identified threats. The mechanisms were grouped according to their functionality, so *"Encryption"*, for instance, could be implemented by any of the identified security mechanisms connected to it (i.e., SSH, SSL, TLS).

The Security Attacks view, supported by the extended Secure Tropos approach, provides an in-depth view of each of the identified threats and their interaction with the rest of the system. For each threat a number of attack methods are identified, each of which targets one or more vulnerabilities of the system. Such vulnerabilities can be identified both by analysing the system's architecture and via specialised vulnerability repositories (e.g. CVE database). The same sources can also be used for identifying security mechanisms which can protect the system against such vulnerabilities. The vulnerabilities exploited by each of the identified threats, and the types of security mechanisms protecting against each of those vulnerabilities, are visualised within the Security Attacks

view diagrams of Fig. 4 and Fig. 5; this critical information is summarised in Tab. 1.

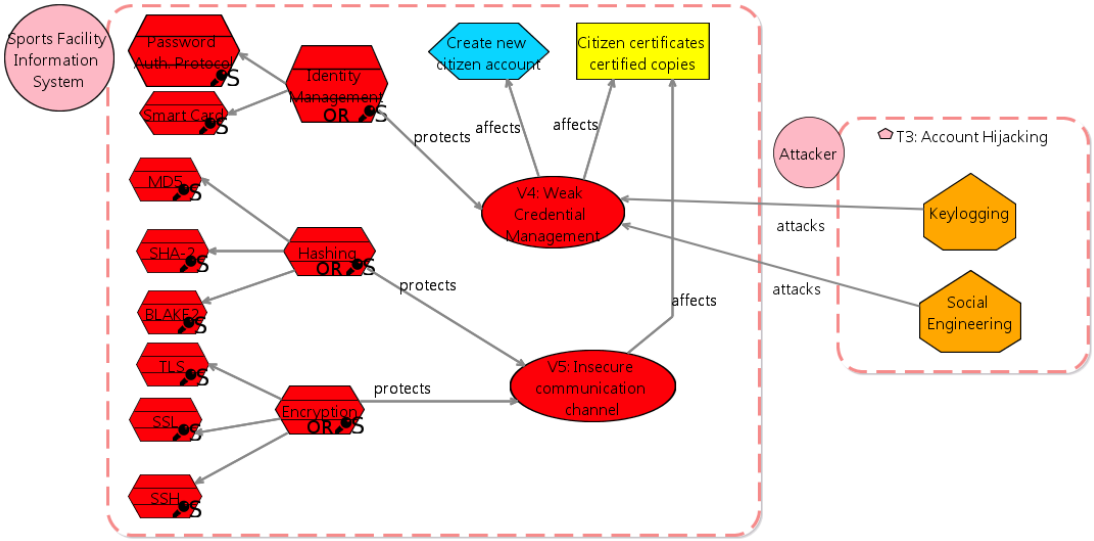


(a) T1: Certificate Data Tampering

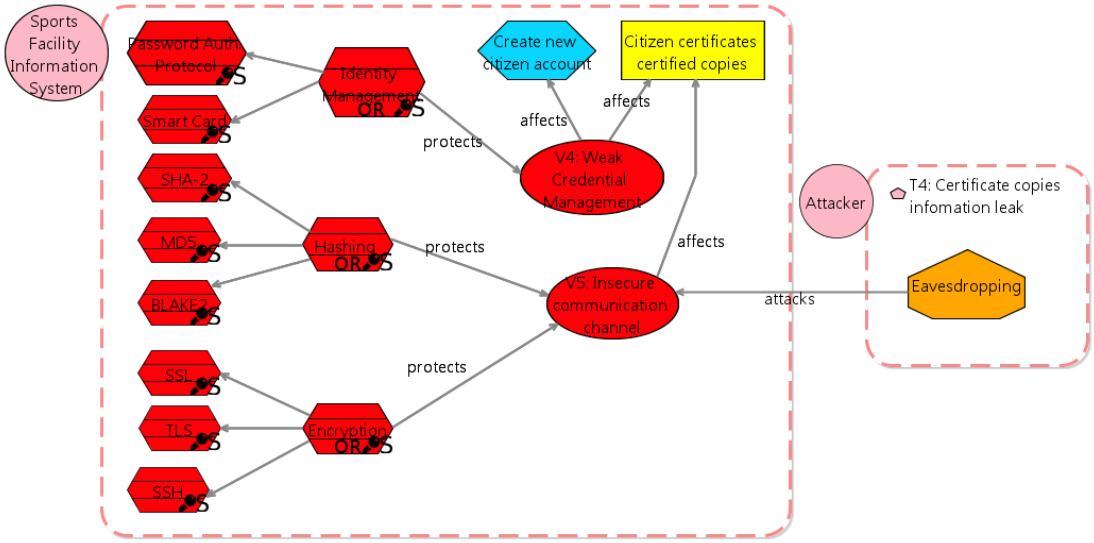


(b) T2: Certificate Information Disclosure

Fig. 4. Security Attacks views of threats T1 and T2



(a) T3: Account Hijacking



(b) T4: Certificate Copies Information Leak

Fig. 5. Security Attacks views of threats T3 and T4

Next, following Steps 2 and 3 of the proposed approach, impact and likelihood values were assigned to the identified vulnerabilities using AHP. These values can be seen in Tab. 2, which also displays the values for the Inherent Risk

Table 1. Matching of Threats, Vulnerabilities and Security Mechanisms

Threat	Vuln.	Encryption	Access Control	Hashing	Id Mgmt (EMACS)	Id Mgmt (SP)
T1	**V1**		✓	✓	✓	
T1, T2	**V2**	✓	✓			
T2	**V3**		✓			
T3	**V4**					✓
T4	**V5**	✓		✓		

Threats: T1: *Certificate Data Tampering*,
T2: *Certificate Information Disclosure*,
T3: *Account Hijacking*,
T4: *Certificate Copies Information Leak*

Vulnerabilities: **V1**: *Insecure Data Handling*,
V2: *Insecure Communication Protocol*,
V3: *Compromised Web Application*,
V4: *Weak Credential Management*,
V5: *Insecure Communication Channel*

(computed as per the instructions in Section 3.2). Then, in order to satisfy the goal model, all of the security constraints, for each agent, must be satisfied. Given that a security constraint is satisfied when at least one of the proposed security mechanisms is proposed and that a security mechanism might satisfy multiple constraints, the final step of our process selects a set of mechanisms to be implemented that satisfies all the security constraints while minimising risk, as defined in Section 3.2.

Table 2. Assignation of Impact and Likelihood values

Threat	Vulnerability	Impact	Likelihood	Inherent Risk
T1	V1	0.15	0.4	0.15
	V2	0.15	0.6	
T2	V2	0.15	0.25	0.2625
	V3	0.3	0.75	
T3	V4	0.25	1	0.25
T4	V5	0.15	1	0.15

A collection of scenarios have been developed and applied to the swimming pool administration system to help to illustrate an application of our approach in practice. Each scenario captures a combination of stakeholders' needs (in the form of priorities) regarding the functional and non-functional properties of the

system to-be. For instance, a scenario may consider the minimisation of the overall cost or the reduction of a specific threat as its top optimisation priority. Such needs are expressed through defined optimisation variables, their thresholds and their prioritisation, when constructing each scenario.

For the identification of the optimal security implementation for each scenario, as dictated by Step 4 of the proposed approach (in Section 3.2), the system is modelled as a constraint goal model and is then used as input to the OptiMathSAT solver. The variables used to define each scenario are the following:

1. the residual risk of each threat (as defined in Section 3.2),
2. the added cost and
3. the added performance overhead of the implementation.

In each scenario these variables can either have a specific hard threshold or they can be set to be minimised (min). Additionally, the minimisation of each variable can be prioritised against the rest of the variables within each scenario. The thresholds and priorities of each variable (where the residual risk of task Tk is denoted $R_{R_{(Tk)}}$), for each scenario considered, are presented in Tab. 3.

Table 3. Variable thresholds (shown as percentages) and any prioritisation (the superscripts) for each scenario (in each column)

Variable \ Scenario	1	2	3	4	5	6
T1 Res. Risk, $R_{R_{(T1)}}$	min	$min^{[1]}$	$min^{[2]}$	$min^{[2]}$	$< 25\%$	$< 50\%$
T2 Res. Risk, $R_{R_{(T2)}}$	min	$min^{[2]}$	$min^{[3]}$	$min^{[3]}$	$< 25\%$	$< 50\%$
T3 Res. Risk, $R_{R_{(T3)}}$	min	$min^{[3]}$	$min^{[4]}$	$min^{[4]}$	min	$< 75\%$
T4 Res. Risk, $R_{R_{(T4)}}$	min	$min^{[4]}$	$min^{[5]}$	$min^{[5]}$	min	$< 50\%$
Added Cost Coverage	min	$min^{[5]}$	$min^{[1]}$	$min^{[6]}$	$min^{[1]}$	$min^{[1]}$
Performance Overhead Coverage	min	$min^{[6]}$	$min^{[6]}$	$min^{[1]}$	$min^{[2]}$	$min^{[2]}$

To obtain values for these variables for each scenario, each of the proposed security mechanisms is instantiated with numerical values regarding the percentage of mitigation it offers for each of the system's vulnerabilities and its contribution towards added system cost and performance. An overview of the values assigned to the mechanisms of the swimming pool administration system is provided in Tab. 4; here, to aid understanding, we use the simplified notation M_{Vk} to denote the mitigation value of the mechanism in the row for vulnerability Vk, in place of the notation M_{jk} from Section 3.2, where the j would refer to the mechanism in the row, and the k to the vulnerability Vk. The resulting security configurations, presented in Tab. 5, is a combination of the proposed mechanisms which optimally satisfies the parameters of each scenario (which were summarised in Tab. 3).

Table 4. Values assigned for the vulnerability mitigation (M_{Vk}), cost and performance effects of the security mechanisms

Mechanism Group	Security Mechanism	M_{V1}	M_{V2}	M_{V3}	M_{V4}	M_{V5}	Cost	Perf.
Encryption	SSH	0	0.6	0	0	0.6	30	30
	SSL	0	0.3	0	0	0.3	20	20
	TLS	0	0.8	0	0	0.8	40	20
Access Control	Firewall	0.3	0.6	0.4	0	0	50	60
	AntiVirus	0.5	0.3	0.3	0	0	40	70
	Firewall & Antivirus	0.7	0.8	0.7	0	0	90	80
Hashing	MD5	0.3	0	0	0	0.3	10	20
	SHA2	0.6	0	0	0	0.6	30	20
	BLAKE2	0.8	0	0	0	0.8	40	20
Ident. Management EMACS	Password	0.3	0	0	0	0	50	50
	Multi-Factor	0.7	0	0	0	0	60	80
Ident. Management SP IS	Password	0	0	0	0.3	0	50	50
	Smart Card	0	0	0.6	0	0	60	30

Scenario 1: The first scenario represents a simple optimisation of all of the system's variables. Therefore, all the variables are set to be minimised, without any assigned priorities between them, as indicated by the values in the second column of Tab. 3 (i.e. the one indicating Scenario 1). The parameters of this scenario represent a case where the system's stakeholders require a system configuration which minimises the cost and added overhead, whilst at the same time keeps the residual risk of all of the potential threats at a minimum level. Such parameters lead to an implementation including, as shown in the second column of Tab. 5, SSL as the selected encryption technology, a firewall as an access control mechanism, MD5 as a hashing algorithm and Password Authentication Protocol as the authentication mechanism of choice for both the EMACS and the Swimming Pool Information System.

Scenario 2: The second scenario presents a variation of the first scenario, emphasising the minimisation of risk from the identified threats. Thus, explicit priorities are set for the optimisation variables, as indicated by their superscript values, shown in the Scenario 2 column of Tab. 3. All variables are still set to be minimised but in this case the optimisation process prioritises the minimisation of the residual risks of the four identified threats in a descending priority, starting with T1 (priorities [1] to [4] in the third column of Tab. 3). The minimisation of the non-functional goals (priorities [5] for cost and [6] for performance) follows in the prioritisation of the optimisation variables. The solution identified in this case, as shown in the second column of Tab. 5, includes, TLS for encryption, both Firewall and Antivirus as access control mechanisms, BLAKE2 for hashing, multi-factor authentication for the EMACS system and Smart Card authentication for the Swimming Pool Information System.

Table 5. Resulting recommendation of system configurations for each given scenario

Scenario / Mech.	1	2	3	4	5	6
Encryption	SSL	TLS	SSL	TLS	TLS	TLS
Access Control	Firewall	Firewall & AntiVirus	AntiVirus	Firewall	Firewall & AntiVirus	Firewall
Hashing	MD5	BLAKE2	MD5	BLAKE2	BLAKE2	MD5
Ident. Mgmt EMACS	Password	Multi-Factor Authent.	Password	Password	Multi-Factor Authent.	Password
Ident. Mgmt SP IS	Password	SmartCard	Password	SmartCard	Password	SmartCard

Scenario 3: The third scenario represents an optimisation process aiming to minimise the cost of the system configuration. As such, all variables are set to be minimised with the cost variable being given the top optimisation priority. The minimisation of the residual risks of all four threats are given the next highest optimisation priority, followed by the added performance overhead, as indicated by the priority values of the fourth column of Tab. 3. The implementation suggested as a result includes security mechanisms with the lowest cost value (i.e., SSL, Antivirus, MD5 and Password Authentication Protocol for both the EMACS and the Swimming Pool Information System).

Scenario 4: The fourth scenario is similar to the third but, the minimisation of the added performance overhead is the main concern of the stakeholders. Therefore, the performance variable has the top optimisation priority, with the residual risks following and the cost being the bottom priority. The identified solution, as shown in the fifth column of Tab. 4, includes security mechanisms adding the least to the performance overhead of the system (i.e., TLS, Firewall, BLAKE2, Password Authentication Protocol and Smart Cards).

Scenario 5: The fifth scenario represents an optimisation process aiming to minimize the cost and performance overheads while also keeping the risk introduced by specific threats below certain levels. Therefore, explicit thresholds have been set for the residual risks of threats 1 and 2 in order to limit them to a maximum of 25% of their initial inherent risk, whilst the added cost and performance overhead have been assigned as the top two priorities for the optimisation process. As a result of the optimisation process, the implementation proposed contains TLS, both Firewall and Antivirus, BLAKE2, multi-factor authentication for the EMACS system and Password Authentication Protocol for the Swimming Pool Information System.

Scenario 6: The final scenario is similar to the fifth but, in this case, maximum accepted risk thresholds have been defined for all of the identified threats. The top optimisation priorities have been assigned to the non-functional aspects of the system (i.e., cost and performance). The maximum accepted values of resid-

ual risk for each threat, as indicated in the last column of Tab. 3, is expressed as a percentage of the initial inherent risk of each threat. The produced solution suggests the combination of TLS, firewall, MD5, Password Authentication Protocol and Smart Cards as the mechanisms of choice.

4.3 Discussion

The capability of our proposal to successfully adapt the system-at-hand within a diverse range of such scenarios, indicates that: (i) the proposed approach is adequately equipped to capture the contextual information necessary for quantitative risk assessment; and (ii) it can be used as the main input for an analysis process which is able to produce appropriate system configurations. The ability of the approach to accommodate any number of variables in the analysis process, both risk and soft-goal related, adds to its flexibility. For instance, in our case study, we identified four potential threats and two qualitative soft-goals for the whole system. However, the application of the approach could easily scale in the case that more threats, soft-goals and security mechanisms were identified. The same applies for the number of different security configurations identified through the presented scenarios. In our case study, we chose six scenarios in each of which the priorities or the maximum accepted values of the involved variables were different. We decided to do so in order to demonstrate the ability of the approach to generate different security configurations under diverse conditions. Nevertheless, any number of scenarios can be generated during the application of this approach, in order to reflect the needs and limitations of the system that is being analysed.

Finally, it is worth indicating that certain aspects of the analysis in this case study were performed as a proof-of-concept and are not meant to be exhaustive. Therefore, a more complete security analysis could be supported by this approach if a more detailed constraint, threat or security mechanism elicitation process takes place with the participation of the security experts. The same is true for the value assignment of the variables related to the impact and likelihood of the identified vulnerabilities and the mitigation, cost and performance overhead of the identified security mechanisms. Nevertheless, the accuracy and completeness of the security analysis presented in the example used for this case study does not adversely affect the capabilities of the proposed approach. In general, reflection on such generated scenarios in conjunction with stakeholders (or even enabling their own experimentation with altering the values), may also help them to deepen their understanding of the implications of the prioritisation and threshold decisions.

5 Related Work

The work of Cailliau and Van Lamsweerde (2012) introduces a probabilistic framework for goal-oriented risk analysis which performs quantitative reasoning

using formal semantics in order to identify the effect of risks on the achievement of system goals. In Chatzikonstantinou et al. (2014), decision task models (DTMs), an extension of goal model diagrams, are introduced, which are able to capture temporal dimensions on goal tree structures, upon the nodes of which cost and benefit values can be attached. Based on such values, and other formally defined constraints, an optimisation process can identify benefit-maximising system compositions. Our approach also makes use of constrained goal models for the performance of trade-off analysis but, in contrast with the above works, it has a clear information security orientation, as it is equipped with concepts and attributes which allows it to measure different aspects of risk and the effects of countermeasures on them.

Elahi and Yu (2007); Elahi and Eric (2011) introduce a security-oriented approach for risk-aware trade off analysis, based on an extension of the i* goal modelling framework. The notation introduced and the tool-supported analysis provided, however, are qualitative and therefore less fine-grained than the one proposed by our approach. A more implementation-oriented approach is presented by Yuan et al. (2013), where architectural patterns are used for performing adaptations to the system according to the results of the evaluation of its security properties during runtime. It does not, however, elaborate on trade-offs between security and other system requirements because it is not meant to be utilised as a design-time approach since it requires a complete system architecture for its application.

The work of Pasquale et al. (2015) introduces a requirements-driven approach for automated and quantitative security trade-off analysis through a sophisticated optimisation algorithm. Similarly, Aydemir et al. (2016) propose a multi-objective, goal-oriented, risk modelling and analysis framework, which is based on constrained goal models and uses OptiMathSAT to identify optimal security countermeasures. As opposed to our approach, the capabilities of these works to capture social aspects of the system are limited. To overcome these limitations, our approaches uses Secure Tropos as the basis of our analysis, after extending it with concepts and attributes that enable the capturing of risk-related aspects. In general, extensions of the Secure Tropos approach can be identified throughout the literature of the area. Such attempts either extend or built on the analysis performed by Secure Tropos in order to support security requirements elicitation in a variety of different contexts. For instance, the work of Argyropoulos, Shei, Kalloniatis, Mouratidis, Delaney, Fish and Gritzalis (2017) utilises Secure Tropos in conjunction with business process modelling for the elicitation of secure service compositions for cloud computing environments. Nevertheless, the focus of the current work is shifted towards the incorporation of risk related aspects of analysis into the existing Secure Tropos framework. A similar attempt towards the alignment of Secure Tropos with the information system security risk management (ISSRM) reference model is presented by Matulevičius et al. (2008) where its risk-related conceptual limitations are identified, the most important of which being the lack of support for expressing and quantitatively evaluating

the concept of risk. The overcoming of such limitations is one of the motivating factors for the development of the approach presented in this work.

6 Conclusions and Future Work

In this paper we introduced an approach built around an extension of Secure Tropos with risk and CGM related concepts. Furthermore, we demonstrated its capabilities of supporting quantitative risk assessment and trade-off analysis between security and other requirements. This was achieved by defining linear cost-functions to estimate the risk of each threat manifestation in our system and optimising various qualitative attributes, such as cost and performance. More specifically, we propose a framework which can use AHP to estimate the likelihood of threats to manifest and the impact of their manifestation on the system under consideration. The prioritisation of the defined cost-functions provided by stakeholders is permitted, and, with the use of an SMT/OMT reasoner, they can be optimising in priority order, and a set of security mechanisms to be used as security countermeasures can be selected accordingly.

This work has established the groundwork for several future directions of research. These include the exploration of the use of alternative reasoners (e.g., Z3 (De Moura and Bjørner, 2008)) which, in contrast with the OptiMathSAT, also support more complex, non-linear cost-functions. Another direction is to try to reduce the framework's current reliance on experts in order to assess the levels of risks and propose security countermeasures, thereby easing the stakeholder's overhead and encouraging adoption. However, some of this reliance is an inherent shortcoming of all risk management approaches, because the definition of quantitative values for risk calculations by experts always introduces a degree of subjectivity. One avenue to pursue to reduce the reliance of the approach on expert input is to explore additional automation techniques for various components. Templates of risk calculation, using impact and likelihood values extracted from online resources and historical data of similar systems, could be provided to users. Such values could then be modified and refined according to the specifics of the system at hand. Thus, users of the approach will not necessarily need to perform the estimation of all values from scratch, since they will be provided with initial suggestions that that can choose to modify. Finally, an interactive scenario-based approach could be used in conjunction with stakeholders to help them to deepen their understanding of the implications of the prioritisation and threshold decisions, and even to potentially revise threshold values or priorities accordingly.

Bibliography

Argyropoulos, N., Alcañiz, L. M., Mouratidis, H., Fish, A., Rosado, D. G., de Guzmán, I. G.-R. and Fernández-Medina, E. (2015), Eliciting security requirements for business processes of legacy systems, *in* 'IFIP Working Conference on The Practice of Enterprise Modeling', Springer, pp. 91–107.

Argyropoulos, N., Angelopoulos, K., Mouratidis, H. and Fish, A. (2017), Decision-making in security requirements engineering with constrained goal models, *in* 'Computer Security: ESORICS 2017 International Workshops, CyberICPS 2017 and SECPRE 2017, Oslo, Norway, September 14-15, 2017', Vol. 10683, Springer, p. 262.

Argyropoulos, N., Shei, S., Kalloniatis, C., Mouratidis, H., Delaney, A., Fish, A. and Gritzalis, S. (2017), A semi-automatic approach for eliciting cloud security and privacy requirements, *in* 'Proceedings of the 50th Hawaii International Conference on System Sciences', pp. 4827–4836.

Aydemir, F. B., Giorgini, P. and Mylopoulos, J. (2016), Multi-objective risk analysis with goal models, *in* 'Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on', IEEE, pp. 1–10.

Blakley, B., McDermott, E. and Geer, D. (2001), Information security is information risk management, *in* 'Proceedings of the 2001 workshop on New security paradigms', ACM, pp. 97–104.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. and Mylopoulos, J. (2004), 'Tropos: An agent-oriented software development methodology', *Autonomous Agents and Multi-Agent Systems* **8**(3), 203–236.

Cailliau, A. and Van Lamsweerde, A. (2012), A probabilistic framework for goal-oriented risk analysis, *in* '2012 20th IEEE International Requirements Engineering Conference (RE)', IEEE, pp. 201–210.

Chatzikonstantinou, G., Athanasopoulos, M. and Kontogiannis, K. (2014), Task specification and reasoning in dynamically altered contexts, *in* 'International Conference on Advanced Information Systems Engineering', Springer, pp. 625–639.

Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. (2000), *Non-functional requirements in software engineering*, Springer.

Dardenne, A., Van Lamsweerde, A. and Fickas, S. (1993), 'Goal-directed requirements acquisition', *Science of computer programming* **20**(1-2), 3–50.

De Moura, L. and Bjørner, N. (2008), Z3: An efficient SMT solver, *in* 'Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems', TACAS'08/ETAPS'08, Springer-Verlag, pp. 337–340.

Elahi, G. and Eric, S. (2011), A semi-automated tool for requirements trade-off analysis., *in* 'CAiSE Forum', Ceur, pp. 9–16.

Elahi, G. and Yu, E. (2007), A goal oriented approach for modeling and analyzing security trade-offs, *in* 'International Conference on Conceptual Modeling', Springer, pp. 375–390.

Giorgini, P., Mylopoulos, J., Nicchiarelli, E. and Sebastiani, R. (2003), 'Formal reasoning techniques for goal models', *J. Data Semantics* **1**(1), 1–20.

Ishizaka, A. and Labib, A. (2009), 'Analytic hierarchy process and expert choice: Benefits and limitations', *Or Insight* **22**(4), 201–220.

Islam, S., Fenz, S., Weippl, E. and Mouratidis, H. (2017), 'A risk management framework for cloud migration decision support', *Journal of Risk and Financial Management* **10**(2), 10.

ISO/IEC (2008), 27005:2008 – Information technology –Security techniques – Information security risk management, Technical report, ISO/IEC.

ISO/IEC (2014), 27000:2014 – Information technology – Security techniques – Information security management systems – Overview and vocabulary, Technical report, ISO/IEC.

Karlsson, J. and Ryan, K. (1997), 'A cost-value approach for prioritizing requirements', *IEEE software* **14**(5), 67–74.

Matulevičius, R., Mayer, N., Mouratidis, H., Dubois, E., Heymans, P. and Genon, N. (2008), Adapting secure tropos for security risk management in the early phases of information systems development, *in* 'International Conference on Advanced Information Systems Engineering', Springer, pp. 541–555.

Mell, P., Scarfone, K. and Romanosky, S. (2007), A complete guide to the common vulnerability scoring system version 2.0, *in* 'FIRST-Forum of Incident Response and Security Teams', pp. 1–23.

MITRE (2017), 'Common attack pattern enumeration and classification, (CAPEC)'.
URL: *https://capec.mitre.org/*

Mouratidis, H., Argyropoulos, N. and Shei, S. (2016), Security requirements engineering for cloud computing: The Secure Tropos approach, *in* 'Domain-Specific Conceptual Modeling, Concepts, Methods and Tools', Springer, pp. 357–380.

Mouratidis, H. and Giorgini, P. (2007), 'Secure Tropos: a security-oriented extension of the Tropos methodology', *International Journal of Software Engineering and Knowledge Engineering* **17**(2), 285–309.

Nguyen, C. M., Sebastiani, R., Giorgini, P. and Mylopoulos, J. (2016), 'Multi-objective reasoning with constrained goal models', *Requirements Engineering* .

Open Web Application Security Project (2015), Application threat modeling, Technical report, OWASP.

Pasquale, L., Spoletini, P., Salehie, M., Cavallaro, L. and Nuseibeh, B. (2015), 'Automating trade-off analysis of security requirements', *Requirements Engineering* pp. 1–24.

Saaty, T. L. (1980), *Analytic hierarchy process*, Wiley Online Library.

Saaty, T. L. (1988), What is the analytic hierarchy process?, *in* 'Mathematical models for decision support', Springer, pp. 109–121.

Sebastiani, R. and Trentin, P. (2015), OptiMathSAT: A tool for optimization modulo theories, *in* 'Computer Aided Verification - 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part I', pp. 447–454.

Stoneburner, G., Goguen, A. and Feringa, A. (2002), Risk management guide for information technology systems (NIST special publication 800-30), Technical report.

Vaidya, O. S. and Kumar, S. (2006), 'Analytic hierarchy process: An overview of applications', *European Journal of Operational Research* **169**(1), 1–29.

Viduto, V., Maple, C., Huang, W. and Bochenkov, A. (2012), A multi-objective genetic algorithm for minimising network security risk and cost, *in* 'High Performance Computing and Simulation (HPCS), 2012 International Conference on', IEEE, pp. 462–467.

Yuan, E., Malek, S., Schmerl, B., Garlan, D. and Gennari, J. (2013), Architecture-based self-protecting software systems, *in* 'Proceedings of the 9th international ACM Sigsoft conference on Quality of software architectures', ACM, pp. 33–42.