Information Security Risk Management

Salimeh Dashti(✉), Paolo Giorgini, and Elda Paja

DISI, University of Trento, Via Sommarive, 14, 38123 Trento, Italy
salimeh.dashti@studenti.unitn.it, {paolo.giorgini,elda.paja}@unitn.it

Abstract. Security breaches on the socio-technical systems organizations depend on cost the latter billions of dollars of losses each year. Although information security is a growing concern, most organizations deploy technical security measures to prevent security attacks, overlooking social and organizational threats and the risks faced because of them. In this paper, we propose a method to information security risk analysis inspired by the ISO27k standard series and based on two state-of-art methods, namely the socio-technical security requirements method STS and the risk analysis method CORAS. The method captures social interactions among stakeholders, while capturing both the risks that threaten their assets as well as those arising while interacting with others. Then, the method suggests how assets are to be protected based on the information classification and potential losses incurred by security breaches. An example from the healthcare domain is used throughout the paper to illustrate the method.

Keywords: Information security · Security risk analysis · Security requirements engineering

1 Introduction

Organizations are increasingly investing on information security to protect informational assets and avoid huge monetary losses [22]. Yet the number of security incidents continues to increase [20]. Evidence suggests that most organizations deploy only technical information security countermeasures, such as encryption of data in transit and intrusion detection systems [10,23]. But organizational systems operate in a socio-technical context where they interact with other systems, humans and organizations by exchanging data, sharing information or outsourcing tasks [18]. As such, they may wreck confidentiality by disclosing information in an unauthorized way, crash the integrity of private data, affect availability by relying on untrusted third parties, etc. Therefore, the design of a secure organizational system cannot be handled with traditional security methods (e.g., [4,6,15,26]) but should rather begin with a thorough analysis of its socio-technical context, thereby considering not only technical attacks, but also social and organizational ones [5,12,18].

Moreover, regulations such as Basel II, the Turnbull report and the Sarbanes-Oxley Act, stress on the need for conducting information security and risk analysis conjointly, since the lack of adequate mechanisms for controlling the flow

G. Poels et al. (Eds.): PoEM 2017, LNBIP 305, pp. 18–33, 2017.
https://doi.org/10.1007/978-3-319-70241-4_2

of information through the organization would incur massive financial costs. Information security is therefore an inseparable and important factor in analyzing risk [2]. Despite many Information Security Risk Management (ISRM) approaches [9,14,29,30] have been proposed, they are mainly for certification purposes and related to specific standards, and do not offer any clear and systematic method.

According to the 2015 PWC report [21], the two main reasons for organizations to fail in risk analysis are: (i) *incomplete risk plans*, and (ii) *ineffective risk prioritization*, that is, focusing on a single-criteria such as financial impact rather than a combination of quantitative and qualitative criteria.

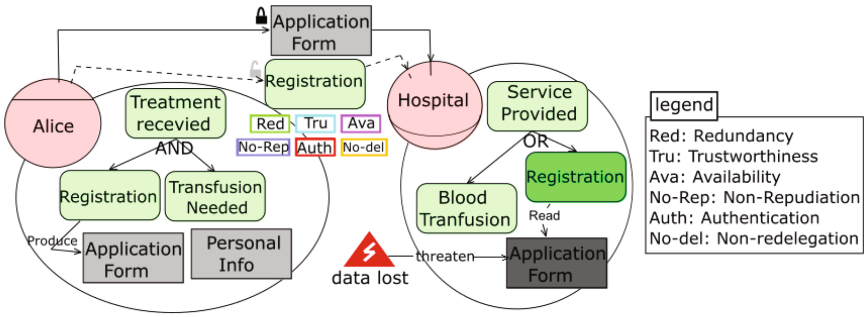
In this paper, we propose an integrated and tool-supported method to information security and risk analysis. We combine and extend two state-of-art methods, namely the socio-technical security requirements method STS [3] and the risk analysis method CORAS [13]. STS is a security requirements engineering method expressly thought for socio-technical systems, it considers information a first-class citizen and deals with security issues arising from social and organizational factors (in particular during interaction). STS allows understanding the impact of threats over stakeholders' assets, but it is not meant thorough risk analysis. Hence, we follow the CORAS method [13], which offers a clear step-by-step method to conduct risk analysis. Following CORAS steps, we address the *incomplete risk plan* problem, by guiding user from identifying risk till treating them, and also we deal with the risk prioritization problem, by analyzing not only the financial impacts of risk but also brand, reputation, and other potential impact factors specific to companies and their industries. However, CORAS does not deal with the risk organizations face over social interactions between stakeholders and clients. By integrating with STS we overcome this limitation. On the other hand, although STS captures security over information, it treats all informational assets equally important from a security point of view. According to the ISO27K security standard series [7,8], information classification is a key concept in the structuring and development of an effective information security method. Thus, we follow the ISO27k principles to specify the classification of a particular informational asset in order to determine how it is to be protected. This helps introducing a balance between protection and costs.

Particularly, our integrated method makes the following contribution: (i) it combines information security and risk analysis providing a systematic method; (ii) it evaluates and classifies assets based on the ISO27k standard series; (iii) it introduces security requirements based on asset classification, and last but not least (iv) it provides a balance between cost and protection.

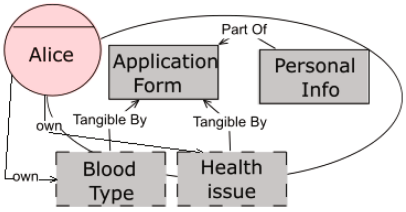
The rest of the paper is organized as follows. Section 2 introduces the baseline and the running example. Section 3 illustrates step by step the modeling phase of our method and how the CORAS steps are placed in our method. Section 4 describes the reasoning techniques and a brief description of the support tool. Section 5 discusses related work. Finally, Sect. 6 describes lessons learned in using the method in the healthcare case study and concludes the paper.

2 Baseline and Running Example

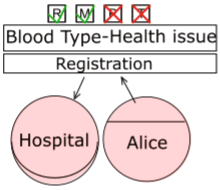
This section introduces the two states-of-art we employed to build our method, and our running example taken from healthcare domain.



(a) Social view.



(b) Information view.



(c) Authorization view.

Fig. 1. The three views of the STS.

STS [3] is a model-based and tool-supported security requirements engineering method for designing socio-technical systems. Models are created based on the Socio-Technical Security modeling language (STS-ml), with STS-Tool[1], which allows constructing models by iteratively building three views (*social*, *information*, and *authorization*), each focused on different aspects of the socio-technical system. The *social view* (Fig. 1a), represents actors as intentional and social entities. STS-ml supports two types of actors: agent–concrete participants (e.g., Alice), and role–abstract actors (e.g., Hospital), used when the actual participant is unknown. Actors may possess documents, which are represented by rectangles (e.g., Application form). Possession indicates that actor has the document and can performing operations and transferring them. The operations are *read*, *modify*, or *produce* documents while achieving their goals, represented by ovals (e.g., Registration). As shown in Fig. 1a, Alice's main goal is to obtain Treatment received which is and-decomposed into two subgoals: Transfusion needed and Registration; to obtain the latter, she *produce* the document Application form.

[1] http://www.sts-tool.eu/.

Threats are represented in terms of *event* in STS. For instance, event data lost has threatened Application form. Security requirements are specified over interactions, namely goal delegations and document transmissions. The locks placed in top-left side of the goal/document indicate that security requirement has been set. Double-clicking on the closed lock opens it and shows the set security requirement which are expressed by small rectangle under asset/goal (e.g., No-Rep, No-del, etc.).

Informational content of the documents manipulated in the social view, are captured in information view (Fig. 1b). The view allows for specifying information ownership (*owns*) which indicates that an actor is the legitimate owner of the asset and can make use of it. For instance, Fig. 1b shows that the agent Alice owns the information Blood type and Health issue, represented by dashed-boarder rectangle. The view also gives a structured representation of information and documents, through *Part-of*, and how they are inter-connected, through *Tangible-by* relation. Figure 1b illustrates that the document Personal info is *part of* the document Application form; the latter represents two pieces of information, namely Blood type and Health issue, via *Tangible-by* relation.

The *authorization view* shows the authorizations that actors grant to others over information, specifying which operations they are allowed (prohibited) to do, for which goals (scope). Plus, specifying whether authorization can be further transferred or not. Figure 1c shows that Alice authorizes Hospital to *read* and *modify* (R and M shown with check sign), but prohibits *transmission* and *producing* (T and P shown with cross sign) information Blood type and Health issue in the scope of the goal Registration. The authorization is transferable since the arrow is solid, while nontransferable authorization is captured by dotted arrow.

CORAS [13] is a model-driven risk analysis method that consists of 8 steps. The first 5 are concerned with the definition of assets and the scope of their analysis. Then, it follows a discussion with stakeholders to order assets accordingly to their relevance. Threats are identified and modeled by using *threat diagram*, as shown in Fig. 2. The CORAS supports three types of threats: *non-human threat* (e.g., System failure), *human-deliberate threat* (e.g., Hacker), and *human-accidental threat* (e.g., Physician). The diagram identifies the vulnerabilities (weaknesses) that opens for, or may be exploited by a threat, to initiate a chain or series of events (threat scenarios) that leads to unwanted incident(s) which harms or reduces the value of an asset(s). As can be seen in Fig. 2, due to the vulnerability Ineffective protection, human-deliberate threat Hacker initiates the threat scenario Breaks in. The scenario leads to the unwanted incident Record theft which impact asset Application form. The frequency of threat scenarios and unwanted incident will be estimated and assigned as *likelihood scale*. Moreover, the impact of incident over asset will be estimated and assigned as *consequence scale*. For instance, the Hacker *certainly* (likelihood scale) exploit the weakness and breaks into system, which has a *Major* (consequence scale) impact on the asset. *Risk evaluation matrix* is used to evaluate risks based on the scales and risk evaluation criteria, which aids the analyst by highlighting risks with green,

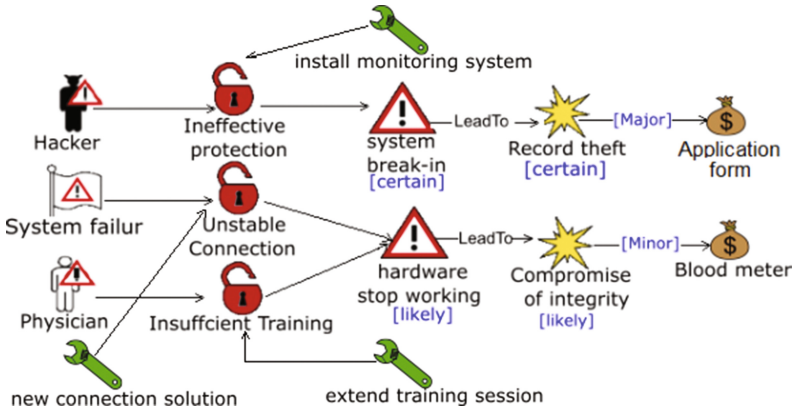


Fig. 2. The CORAS threat diagram. (Color figure online)

orange and red colors; the threats that fall in the range of cells highlighted in red cells will be treated. In the example shown in Fig. 2 all the captured threats are treated. For example, the suggested treatment to avoid harm by Physicion is to conduct training sessions.

Running Example. A healthcare system is a socio-technical system in which hospitals and healthcare centers allow physicians or general practitioners to perform medical tests and give advice to patients who have registered for medical services. Such a scenario sees as main participants not only the hospitals, patients, and physicians, but also laboratories for specialized tests as well as research centers that conduct data analysis to make forecasts on the need for blood banks. This is complex socio-technical system in which involved participants (actors) need to rely on each other to fulfill their objectives, by interacting and exchanging information. Information in such system is sensitive, for instance personal information, health status, etc., which should be protected from possible attacks. A threat is not necessarily an outsider, like a hacker, it also can be a careless employee or a connection failure. This example integrates two scenarios from the healthcare used to apply CORAS and STS, the methods that serve as a baseline for our work. As such, the illustration of the integrated method can show of the added values of the latter compared to the underlying methods.

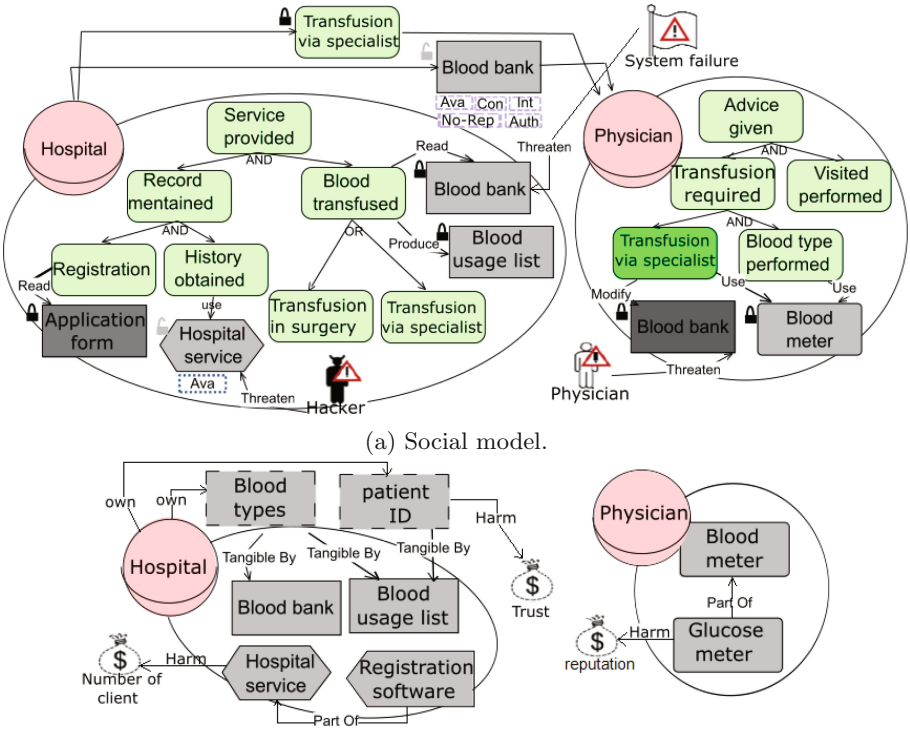
3 Information Security Risk Modeling

Our method consists of two main macro phases: *modeling* and *automated analysis*. In this section, we focus on the modeling phase, while automated analysis and also the reasoning techniques and the supporting tool will be described in the next section.

The modeling phase consists in the creation of four different models, each focused on a specific aspect of socio-technical system. We start with the *social*

model and continue with describing *asset* and *authorization modeling*. Then, we conclude with *threat modeling*.

1. Social Modeling. This model represents the organization of the overall socio-technical system. Figure 3a represents part of the *Social Model* of our example. We have extended the concept of asset in STS by adding three different types of assets: (1) *hardware*, which is represented by round-corner rectangle (e.g., Blood meter); (2) *software*, which is represented by pentagon (e.g., Registration software), and (3) *system*, which is represented by hexagon (e.g., Hospital service). Actors *use* these assets while achieving their goals. Figure 3a shows that Physician *uses* hardware Blood meter to achieve two subgoals: Blood type performed and Transfusion via specialist.



(a) Social model.

(b) Asset model.

Fig. 3. Part of the models of the running example.

STS supports clause A.6.2.3 of ISO27002 and control A.6.2.1 of ISO27001 which require covering security requirements on agreements with third parties. We have gone further by acknowledging the importance of protecting assets *within organizations*, with respect to control A.6.1 of ISO27001 and Clause 6.1 of ISO27002. For instance, role Hospital wants to ensure the availability of the

service Hospital service to achieve subgoal History obtained. Thus, security requirements availability (Ava) has been set.

Another extensions to this model is to identify threats in a technical way by specifying the type of threats, that is, to employ proper and better treatment. Consider that careless employee is the threat, as they remained the most cited source of compromise [20], the first treatment could be training him instead of implementing other expensive security countermeasures. Thus, the STS event concept has been refined to the three types of threats (refer to Sect. 2), from CORAS. Figure 3a shows, threat System failure threaten document Blood bank.

2. Asset Modeling. As mentioned, we have added three more information assets which their structures also need to be modeled, as document and information, via part-of and tangible-by relationships (Sect. 2). Figure 3b represents that role Physician possesses two hardwares, namely Glucose meter and Blood meter. The former is *part of* the latter. Also, Hospital posses software Registration software which is part of system Hospital service.

Taken from CORAS, we group assets in two categories, namely *direct asset* and *indirect asset*. The former refers to assets which can be directly attacked (e.g., a server, a document), while the latter refers to what can be harmed only through direct assets (e.g., fame, trust). Figure 3b represents that any harm to information Patient ID results harm to Trust.

As mentioned, the key concept to develop an effective information security is to classify assets based on which security requirements are set. Control A.7.2 of ISO 27001 requires a procedure for classifying assets to ensure an appropriate level of protection, and control A.7.2.1 provides guidelines on classification which is expanded later from control 7.2.1. of ISO27002. Our method follows the steps offered by ISO27001 to classify assets:

Step 1. Inventory of assets (control A.7.1.1): requires to have a list of asset, which has been covered by means of *asset model*.

Step 2. Asset classification (control A.7.2.1): aims at valuing adverse effect of loss of security objectives (Confidentiality, Integrity and Availability) in case of security breaches on organization. We define three classes which represent three levels of adverse effects: (1) *low* for limited harm; (2) *moderate* for serious harm, and (3) *high* for catastrophic harm. We assign a value from 1 to 3, for low to high. The given value to each security objectives determine the level of harm that compromising them will lead to. Therefore, how each of the security objectives should be protected. The process of valuation is called *Asset Evaluation*. With this approach, we suggest to protect assets that are valuable to their owner, even if there is no captured threat against them; that is because the ways that are impossible to attack system yesterday may get possible tomorrow as technology is growing dramatically.

Asset evaluation has to be done over meetings with stakeholders. Understanding the impact of indirect assets on organization, is a starting point to evaluate direct assets. Thus, the process begins with valuing indirect assets, from 1 (low) to 3 (high). As shown in Fig. 4, role Hospital has evaluated the adverse effect of

Name	Blood usage list	Value	C	I	A	Type	Document
			2	2	2		
Number of user	1		Number of copy			1	

Direct Asset							Indirect asset	
Name	Num. of user	Num. of copy	type	C	I	A	Name	Value
Blood type	4	1	Info	2	2	2		
Patient ID	1	2	Info	1	2	2	Trust	2

Fig. 4. Asset valuation table

Trust, *moderate* as valued 2. After valuing indirect assets, we need to value direct assets, for which asset structure is so helpful, since value of an asset is defined by value of what it is composed of. Subsequently, we start from asset constituent by answering questions such as: *"how much harm will the unauthorized disclosure of this asset cause to the organization?"* to evaluate confidentiality. Same sort of question to evaluate all security objectives. Once evaluation of all constituents is done, the highest given value to security objectives among constituents of an asset, will be automatically assigned to corresponding security objectives of that asset.

Values are set, using Asset Valuation Table, shown in Fig. 4. The figure depicts that document Blood usage list represents information Blood type and Patient ID, which are valued 2, 2, 2 and 1, 2, 2 for CIA, respectively. Thus, the automatic assigned value to confidentiality is 2, since it is the highest among 1 and 2; same for integrity and availability.

By considering indirect assets while valuing the loss of organization in case of harm to their direct asset, we offer a multi-criteria evaluation which addresses the mentioned "single-criteria" problem from [21].

Step 3. Asset handling (control A.7.2.2): We follow STS's principle [3] in classifying security requirements. Due to space limitation, in the following we describe only the added security requirements type to STS, for the rest please refer to [3]:

- ***Confidentiality:*** where we introduce:(i) Number of copies (ISO27002 clause 6.2.3): to restrict the number of instances that can exist from information to avoid disclosing information; (ii) Number of users (ISO27002): to control number of permitted users to access information asset; (iii) Duration of authorization (ISO27002 clause 6.2.3), and (iv) Act on termination (ISO27001 control A.8.3.1 and A.8.3.2). The last two requirements allow capturing access control when giving authorization to another party which are introduced during *Authorization Modeling*. While, requirements *Number of copy* and *Number of user* are set in this model by Asset Valuation Table (Fig. 4).
  The figure shows that, *Number of user* and *Number of copy* are 4 and 1 for information Blood type and 1 and 2 for information Patient ID, respectively.

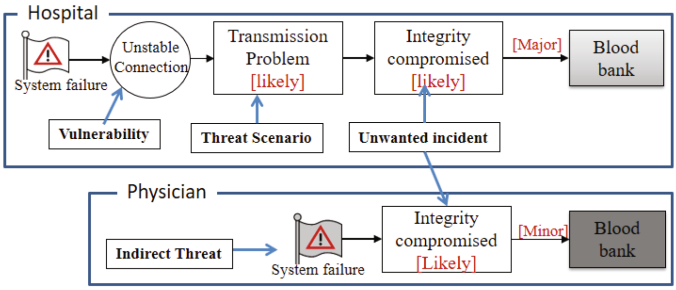
Unlike value of CIA, the lowest given value to these security requirements among constituents will be automatically assigned to corresponding security requirements of their main asset. The reason is that, if it is required to have one instance from a piece of information, surly there should be only one document which contain it. Otherwise, the security requirement for that piece of information is violated. The same goes for *number of user*. Accordingly, the requirements for document Blood usage list are 1 and 1.

– ***Accountability*** refined as: (i) *Non-repudiation of transmission*: expressed by receiver who requires sender not to repudiate the transmission of asset; (ii) *Non-repudiation of acceptance*: expressed by sender who requires receiver not to repudiate receiving transmitted asset. Figure 3a shows the requirements over transmission of document Blood bank. (iii) *Separation of information* (based on effect of aggregation concept by ISO27002): aggregation of information may cause a large quantity of non-sensitive information to become sensitive. This security requirement can be set among information which their aggregation makes them sensitive. For instance, information health issue might not be sensitive as long as it is anonymous. Once it appears with the patient name, then confidentiality of it may become important.

As said, the classification given to a particular information asset determines how it is to be protected. We introduce three different levels of security requirements for each level of value: (1) non-negotiable: refers to the security objective valued 3. Such requirement has to be implemented otherwise the company face a severe harm. (ii) negotiable: refers to the security objective valued 2. These requirements may or may not be implemented. Stakeholders and the risk analyst can discuss over it and decide based on the likelihood of captured threat. (iii) No protection: refers to the security objective valued 1 that will pose any harm to organization. If confidentiality of an asset is 3, while integrity and availability are 2, the method requires *non-negotiable* security requirements for confidentiality and *negotiable* for the other two security objectives, so that, we invest more where needed to make a balance between protection and cost. Note that, the requirements are automatically assigned as user enter the values. Graphically, non-negotiable security requirement is expressed by dashed-line border and negotiable ones by dotted-line border. As shown in Fig. 4, document Blood usage list has been valued 2, 2, 2. Therefore, *negotiable* security requirements should be assigned, as shown in Fig. 3a.

3. Authorization Modeling. STS authorization model has been extended to support two security requirements: (1) *duration of authorization*: to specify how long the given authorization is valid; and (2) *act on termination*: to determine the proper action once the authorization is over that could be either to *return* or to *destroy* the asset.

By now, the first fifth steps of CORAS are taken by identifying target of analysis (goals and assets) and threats using *social* and *asset* model. Next, we cover step six (risk estimation) and seven (risk evaluation). To do so, we need to model captured threats.



(a) Threat Mode

Blood Bank	Consequence	Insignificant	Minor	moderate	Major	catastrophic	
Likelihood	Rare						Very High
	Unlikely						High
	Possible						Low
	Likely				System failure-Integrity		Very low
	Certain						

(b) Risk Evaluation Matrix of Blood bank.

Fig. 5. Threat modeling.

4. Threat Modeling. To model threat, we use the CORAS concept but fault tree analysis [27] notations which are widely adopted for risk analysis. Figure 5a illustrates that, due to Unstable Connection (vulnerability), it is *likely* (likelihood scale) that the document Blood bank faces Transmission problem (threat scenario); the problem can have a *major* (the consequence scale) impact on the document Integrity (unwanted incident).

Threats can propagate via interactions, namely goal delegation and asset transmission. As shown in Fig. 6a, a non-human threat System failure attacks document Blood bank which has been transmitted to the role Physician to be modified while achieving subgoal Transfusion via specialist. Thus, the compromised document can have side effect not only on functionality of Hospital, but also on the Physician. These types of threats are called *indirect threats*. As Fig. 5a shows, the indirect threat System failure (illustrated in gray) has *minor* impact on Physician, while it has *major* impact on the Hospital. Note that, modeling *indirect threat* only requires *unwanted incident*, since the vulnerability existed where the threat has raised and carried out the events.

To evaluate threats, we use *Asset Evaluation Matrix* (Fig. 5b), from CORAS. The matrix will be automatically filled for each victim asset, based on assigned *likelihood* and *consequence* scales. The threats that fall in the range of the cells highlighted in red need to be treated. The matrix cell can be modified to adjust for each asset based on the risk tolerance of stakeholder (please refer to [13]). Figure 5b illustrates that the threat *System failure* against document Blood bank is unacceptable.

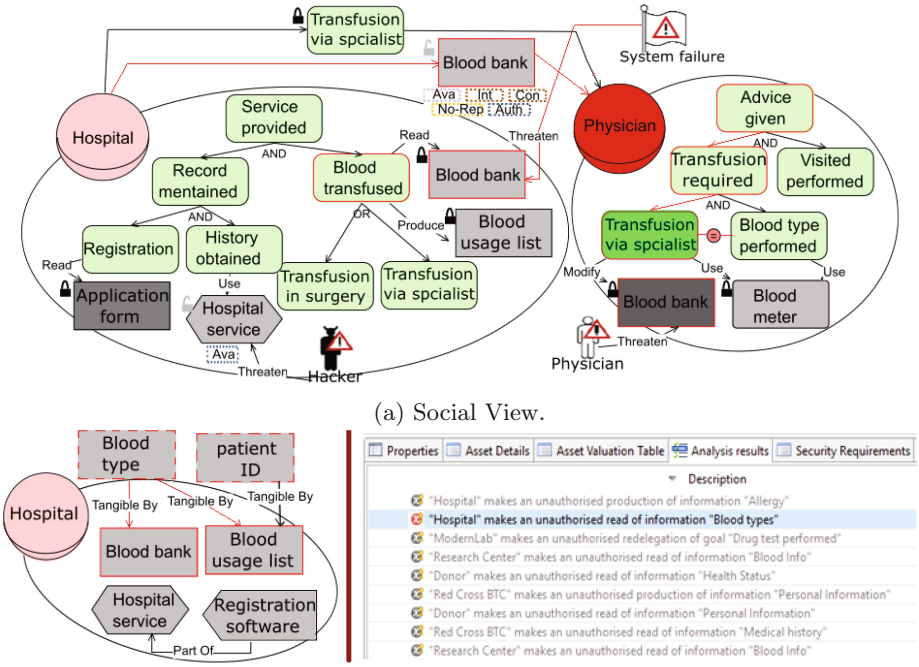
4 Automated Analysis and Supporting Tool

Although modeling languages are useful means to represent knowledge, they might become inconsistent as models grow in size. STS automated reasoning techniques come to help in identifying potential inconsistencies. The analysis are performed based on a formal framework, described in [3]. The method supports three types of analysis: (1) approving the assigned security requirements as treatments (2) verifying that all security requirements can be satisfied, and (3) verifying the impact of threats threatening assets. The first analysis is known as risk analysis, the second as security analysis, while the latter as threat analysis.

(I) Risk Analysis. Risk estimation and evaluation has been covered by *threat modeling*. Treating threats is the last step of CORAS to take.

Assets are already protected by setting security requirements based on their values and any extra security requirements asked by stakeholders (refer to [3]). Yet, after evaluating threats, it needs further checks. The *Non-negotiable* security requirements assigned to assets ought to be implemented, whether the assets are attacked or not. Whereas, *negotiable* security requirements can be further discussed to be or not to be implemented based on frequency of threats. As the result of risk analysis, the tool highlights in red the assets which are under unacceptable attack with negotiable security requirements, so that, stakeholders can decide whether to keep the protection as they are or improve them to non-negotiable type. As shown in Fig. 5b, the threat against the document Blood bank, is unacceptable and we also have shown in Fig. 3a that its security requirements are negotiable. The decision is to protect document Blood bank, strictly. So that, its security requirement has been improved. Graphically, Fig. 6a shows that the security requirements of the document became dotted-border.

(II) Security Analysis. STS supports: (i) identifying possible conflicts among security requirements, and (ii) identifying conflicts between actors own business policies and the security requirements imposed on them. Figure 6c depicts part of this analysis. We extended STS analysis to check the following conflicts over assets as well: (1) fulfillment of security requirement *number of copy*. For instance, the requirements for information Blood type is 1 (Fig. 4). While, as shown in Fig. 6b, the information is represented in two documents, namely Blood usage list and Blood bank. Thus, the requirement is violated, so that, the tool is highlighted them. (2) fulfillment of security requirement *number of user*. This requirements for hardware Blood meter is set one. Figure 6a shows that, the hardware need to be accessed for achieving two subgoals, namely blood type performed and transfusion via specialist, which may lead to conflict if they be achieved by two different user. To avoid such conflicts, we use security requirement *Goal-based combination of duties* between two goals. This requirement implies the fact that the user who fulfills the former, ought to be the same as the one who fulfills the latter. Graphically, this is represented as an arrow between two entities annotated with the "equal" (=) symbol. In our example, as the result of security analysis, *Goal-based combination of duties* between the two mentioned subgoals is suggested, to ensure that the same Physician will achieve both goals.



(a) Social View.

(b) Asset View.

(c) Part of Security Analysis Result.

Fig. 6. Automated analysis result. (Color figure online)

(III) Threat Analysis. This analysis identifies the followings: (1) the threat propagation through the model. Figure 6a, illustrate the result of threat System failure propagation. Earlier we described how we take care of the indirect threat. (2) Specifying critical and non-critical actors. There are two groups of actors in a socio-technical systems: (i) actors who hold assets (even one) with *negotiable* type of protection, called *non-critical actors*; (ii) actors who *only* hold asset with non-negotiable security requirements, called *critical actors*. The entire socio-technical systems can be attacked through one single entry. Thus, the latter group may be more concerned about the level of protection of actors they interact with. Distinguish between these two groups, help stakeholders to either emphasizing strictly on security while interacting with uncritical ones or possibly avoid interactions with them. In our example the role Physician is a critical actor who receives document from a non-critical role. Graphically, the role is highlighted in red, as shown in Fig. 6a.

The Supporting Tool. Our method is fully supported by a prototype CASE tool. It has been developed as an eclipse plug-in of the STS-Tool, which is a modeling tool for STS-ml. It is a standalone application written in Java, and its core is based on Eclipse RCP Engine. AS STS-Tool, our tool is compatible for multiple platforms (Windows 32 and 64 bits, Mac OS X, Linux). The tool

supports all activities of our method providing graphically facilities for modeling and algorithms for automated analysis over models. Moreover, a number of functionalities for report generation is implemented. Once the analyst has developed all models and performed the required analysis, the tool generated automatically a set of documents as support documentation for the work done. The tool is still under implementation and can be found soon in[2].

5 Related Work

We can differentiate two areas related to our work: *information security risk management* and *information security management systems*.

Information Security Risk Management (ISRM) methods are mainly focused on risks but suffer from several issues: lack of a methodology with clear steps, overlooking information security risks, expensive documentation, and the need for a deep understanding and expertise to apply the proposed approaches. In some cases, like *Dutch A&K analysis, Austrian IT security handbook, MARION, ISAMM*, the information is only available in the local language. These issues have been tackled by some works: The MAGERIT risk analysis and management method identifies and groups assets according to their organizational hierarchy. Then, it analyzes potential threats and required safeguards to meet security objectives. It aims to make stakeholders aware of the existence of risks and keep them under control. In a similar way, SREP (Security Requirement Engineering Process) [16] is an asset-based iterative and incremental process that uses misuse case diagrams to model threats and MAGERIT tables to assess them. Despite their systematic nature, both methods overlook vulnerabilities derived from interactions, whether from the system and its environment or from social interactions among stakeholders. IRAM [31] is a workshop-based and tool-supported model, focused on the organization's information systems and information threats. The approach helps determine the criticality and prominence of information systems. Unfortunately, the actual risk calculation formula is not openly available. Mehari [17] provides a complex process, including cyclic risk management and a knowledge base to support semi-automated risk analysis based on a set of input factors. While the method supports quantitative, scenario-based analysis of risk, it lacks the identification of organizational assets, valuing them, and capturing threats against them. Finally, The Facilitated Risk Assessment Process (FRAP) [19] aims to sketch how a "facilitator-led" qualitative risk analysis and assessment can be applied in order to enable stakeholders to produce findings which are understandable by non-experts. However, FRAP strongly relies on the role of the Facilitator to guide the stakeholder, it does not valuate the asset and presents the same drawbacks as MAGERIT and SREP regarding interactions.

Information Security Management Systems (ISMS) is focused on standards for IT Governance which lead to information security, such as PRINCE2, OPM3, MMI, P-CMM, PMMM, ISO27K series, BS7799, PCIDSS, COSO, SOA, ITIL

[2] http://www.sts-tool.eu/downloads/.

and COBIT. In the following, we discuss the five most prominent ISMS standards. *BS7799* [28] contains several parts, the first part containing best practices for ISMS, whereas the second part focuses on how to implement ISMS referring to BS 7799-2, which later became ISO 27001 [7]. The Control Objectives for Information and related Technology (*COBIT*) [11] is a certification which is globally accepted to ensure that IT operations are aligned with business goals and objectives. However, [24] reveals that there is relatively little academic literature making use of COBIT, and [25] claims that a big effort is required to understand and apply it. The Information Technology Infrastructure Library (ITIL) [1] is an approach to service management based on "do what works". It unites all areas of IT service provision into a single goal based on two main concepts of a service: (i) delivering value and (ii) not caring how a service is implemented. However, its monolithic analysis does not capture which actors in the organization generate the values that are later delivered to the customers. Finally The Payment Card Industry Data Security Standard (PCIDSS) [32] is a worldwide information security standard defined to help industry organizations processes card payments.

6 Conclusion

In this paper, we have presented an integrated method for information security and risk analysis built on top of the STS security methods and the CORAS risk analysis method. We adopted principles from the ISO27k series and provided a method to evaluate information-related assets based on their potential impacts in case of security breaches, to classify them and ensure an adequate level of security according to their value. We also find weak actors of the system to warn the analyst to limit interaction with them.

The running example used throughout the paper is part of a larger case study we developed[3]. The main findings while developing the running example are summarized as follows: (1) although the asset classification has provided a number of advantages, grouping them in three categories was in many cases too restrictive. As future work, we will support a customizable classification, so that, based on the socio-technical system under analysis, a different classification will be adopted; (2) the method performs in a high level of abstraction, which provides a rationale on how the business analyst should decide upon security requirements. This makes the estimation of costs difficult. For a more accurate evaluation, we plan to extend our method by estimating the costs of asset acquisition and security requirements implementation; (3) although our method enriched STS by modeling and analyzing information-related assets, the language was not expressive enough to represent how assets are manipulated to achieve goals.

Acknowledgments. This project has received funding from the SESAR Joint Undertaking under grant agreement No. 699306 under European Union's Horizon 2020 research and innovation program.

[3] http://disi.unitn.it/~pgiorgio/IRM-HealthCareCase.pdf.

References

1. Abid, M.: Information technology infrastructure library (ITIL). JIT **1**(1) (2012)
2. Continuity Central: Information risk co-existance. https://www.continuitycentral.com/feature0189.htm
3. Dalpiaz, F., Paja, E., Giorgini, P.: Security Requirements Engineering: Designing Secure Socio-Technical Systems. MIT Press, Cambridge (2016)
4. Firesmith, D.: Security use cases. J. Object Technol. **2**(3), 53–64 (2003)
5. Giorgini, P., Massacci, F., Mylopoulos, J.: Requirement engineering meets security: a case study on modelling secure electronic transactions by VISA and Mastercard. In: Song, I.-Y., Liddle, S.W., Ling, T.-W., Scheuermann, P. (eds.) ER 2003. LNCS, vol. 2813, pp. 263–276. Springer, Heidelberg (2003). doi:10.1007/978-3-540-39648-2_22
6. Haley, C., Laney, R., Moffett, J., Nuseibeh, B.: Security requirements engineering: a framework for representation and analysis. IEEE Trans. Softw. Eng. **34**(1), 133–153 (2008)
7. ISO/IEC: ISO/IEC 27001 information security standard (2013)
8. ISO/IEC: ISO/IEC 27002 information security standard (2013)
9. ISO/IEC 15408: ISO/IEC 15408–1:2009 information technology standard (2009)
10. Kessel, P.: Into the cloud, out of the fog: EY 2011 global information security survey (2011)
11. Lainhart IV, J.W.: A method for controlling information and information technology risks and vulnerabilities. JIS **14**(s-1), 21–25 (2000)
12. Liu, L., Yu, E., Mylopoulos, J.: Security and privacy requirements analysis within a social setting. In: Proceedings of RE Conference, pp. 151–161. IEEE (2003)
13. Lund, M., Solhaug, B., Stølen, K.: Model-Driven Risk Analysis: The CORAS Approach. Springer Science & Business Media, Heidelberg (2010)
14. McEvoy, N., Whitcombe, A.: Structured risk analysis. In: Davida, G., Frankel, Y., Rees, O. (eds.) InfraSec 2002. LNCS, vol. 2437, pp. 88–103. Springer, Heidelberg (2002). doi:10.1007/3-540-45831-X_7
15. Mead, N., Stehney, T.: Security Quality Requirements Engineering (SQUARE) Methodology, vol. 30. ACM, New York (2005)
16. Mellado, D., Fernández-Medina, E., Piattini, M.: Applying a security requirements engineering process. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) ESORICS 2006. LNCS, vol. 4189, pp. 192–206. Springer, Heidelberg (2006). doi:10.1007/11863908_13
17. Mihailescu, V.: Mehari. JABIS **3**(4), 143 (2012)
18. Paja, E., Dalpiaz, F., Giorgini, P.: Modelling and reasoning about security requirements in socio-technical systems. Data Knowl. Eng. **98**, 123–143 (2015)
19. Peltier, T.: Facilitated risk analysis process (FRAP). Auerbach Publication, CRC Press LLC, Boca Raton (2000)
20. PWC: Turnaround and transformation in cybersecurity. https://www.pwc.com/sg/en/publications/assets/pwc-global-state-of-information-security-survay-2016.pdf
21. PWC: Why risk assessments fail. https://www.pwc.com/us/en/risk-assurance/publications/assets/preventing-erm-risk-assessment-failure.pdf
22. PWC: 2015 information security breaches survey (2015). https://www.pwc.co.uk/assets/pdf/2015-isbs-technical-report-blue-digital.pdf
23. Richardson, R.: CSI computer crime and security survey, 2011 (2010)

24. Ridley, G., Young, J., Carroll, P.: COBIT and its utilization: a framework from the literature. In: Proceedings of the 37th HICSS. IEEE (2004)
25. Simonsson, M., Johnson, P., Wijkström, H.: Model-based it governance maturity assessments with COBIT. In: ECIS, pp. 1276–1287 (2007)
26. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requirements Eng. **10**(1), 34–44 (2005)
27. Stamatelatos, M., Vesely, W., Dugan, J., Fragola, J., Minarick, J., Railsback, J.: Fault Tree Handbook (2002)
28. British Standard: Code of practice for ISM British Standards Institution (1995)
29. British Standard: Standard 100-2: It-grundschutz-vorgehensweise. BSI (2008)
30. Stoneburner, G., Goguen, A.Y., Feringa, A.: SP 800-30. Risk management guide for information technology systems (2002)
31. Sun, L., Srivastava, R., Mock, T.: An information systems security risk assessment model under the Dempster-Shafer theory of belief functions. JMIS **22**(4), 109–142 (2006)
32. Virtue, T.: Payment Card Industry Data Security Standard Handbook. Wiley, Hoboken (2009)

http://www.springer.com/978-3-319-70240-7

The Practice of Enterprise Modeling
10th IFIP WG 8.1. Working Conference, PoEM 2017,
Leuven, Belgium, November 22-24, 2017, Proceedings
Poels, G.; Gailly, F.; Serral Asensio, E.; Snoeck, M. (Eds.)
2017, XIII, 363 p. 99 illus., Softcover
ISBN: 978-3-319-70240-7