Pattern-Based Security Requirements Derivation from Secure Tropos Models

Atilio Rrenja, Raimundas Matulevičius

Pattern-based Security Requirements Derivation from Secure Tropos Models

Atilio Rrenja and Raimundas Matulevičius

Institute of Computer Science, University of Tartu, Estonia
{rrenja, rma}@ut.ee

Abstract. The increasing rates of cyber-attacks have led to the subsequent need to rapidly develop secure information systems (IS). Secure Tropos is an actor and goal-oriented approach to identify security goals and to enable security requirements elicitation. This is achieved by considering system actors, their dependencies and by deriving security constraints that actors need to satisfy. Nevertheless goal-oriented modelling has proven itself to be valid it also contains few shortcomings. One of them is the high granularity of the process, which leads quickly to high complexity models. Security patterns are proven to be reusable solutions that address recurring security problems. In this paper we investigate the integration of a pattern-based security requirements derivation from the Secure Tropos models.

Keywords: Security risk management, Secure Tropos, Security patterns.

1 Introduction

Security concerns play an important role in nowadays enterprises. Different enterprise stakeholders have various objectives and need to collaborate to achieve them. Thus, understanding security risks and estimating their impact could envision threats, estimate their consequences, and propose countermeasures to mitigate these threats.

Secure Tropos is an agent-oriented information and enterprise system development method that helps understanding *security objectives* through *satisfying security constraints* by considering *actor dependencies* [11]. In [7] [8] Security Tropos was extended to Security Risk-aware Secure Tropos (RAST), where the original language was semantically aligned to the concepts of the domain model for information systems security risk management (ISSRM) [6] [9]. The extended language supports security requirements elicitation through understanding security risks. However, even given an IS with a rather moderate complexity, identifying and mitigating security risks could become quite a complex activity. One of the reasons is the inherited complexity of the Secure Tropos model, when the model size quickly grows with introduction of different analysis concerns.

In this paper we propose an application of security risk-oriented patterns (SRPs) [2], which could overcome the above problem by suggesting the proven security solutions for the reoccurring security problems. We analyse *how to apply SRPs and derive security requirements from Secure Tropos models*. To answer the question, firstly, we have represented SRPs using RAST. Secondly, we have proposed a process

to apply SRPs to derive security requirements from the (Secure) Tropos models. Finally, we have conducted an observatory study to understand usability of the proposed method.

The rest of the paper is structured as follows: in Section 2 an overview of security risk management using Secure Tropos is provided. Section 3 presents security risk-oriented patterns. In Section 4 we consider the process for security requirements derivation from the Secure Tropos model. Section 5 outlines the observatory study conducted in order to validate the usability and understandability of the pattern application. Section 6 discusses some related work. Finally in Section 6 we summarise the study discussion and present some future work.

2 Security Risk Management using Secure Tropos

In this section we, firstly, present the ISSRM domain model used to define the SRPs and to analyse the Secure Tropos models. Secondly, we overview how Secure Tropos constructs are aligned to concepts of the ISSRM domain model.

2.1 Information Systems Security Risk Management

The ISSRM domain model (see Fig. 1) defines security risk management concepts at three interrelated levels, which help developers identify specific IS security risk management constructs [6] [9].

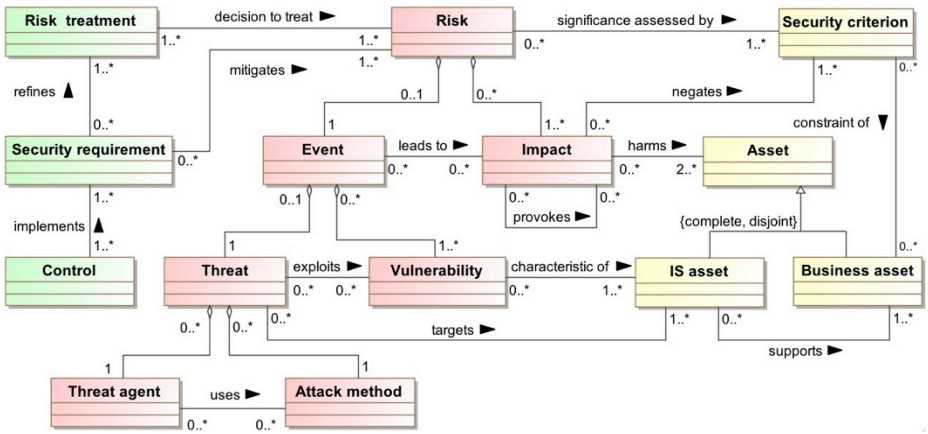

Fig. 1. The ISSRM Domain Model (adapted from [6] [9])

Asset-related concepts (i.e., *business* and *IS assets*, and *security criterion*) explain the organisation's values that need to be protected. The needed protection level is defined as the security needs, typically in terms of confidentiality, availability and integrity. *Risk-related concepts* (i.e., *risk*, *impact*, *event*, *vulnerability*, *threat*, *attack method*, and *threat agent*) define the risk itself and its components. Risk is a combination of threat with one or more vulnerabilities, which leads to a negative

impact, harming some assets. An impact shows the negative consequence of a risk on an asset if the threat is accomplished. A vulnerability is a weakness or flaw of one or more IS assets. An attack method is a standard means by which a threat agent executes a threat. *Risk treatment-related concepts* (i.e., *risk treatment decision*, *security requirement* and *control*) describe how to treat the identified risks. A risk treatment leads to security requirements mitigating the risk, implemented as security controls.

The risk management process consists of six steps. First, it initiated by identifying analysed context and assets. The second step is security objective determination. The third step includes risk analysis. This step is followed with making a risk treatment decision. In the fifth step one suggests security requirements, which are implemented to security controls (sixth steps). The process is iterative and each previous step could be repeated if its result is not of satisfactory quality.

2.2 Security Risk-aware Secure Tropos

Security Risk-aware Secure Tropos (RAST) is an extension of the *i** framework [13], Tropos [4] and Secure Tropos methods [11]. By aligning the modelling constructs to the concepts of the ISSRM domain model, it becomes possible to use the targeted modelling constructs to express specific concepts from the security risk management domain. This extension enables using Secure *i**/Tropos concepts wherever possible utilizing the already existing constructs, but additionally, whenever void or ambiguity exits, new constructs are introduced to address security risk management.

Asset-related concepts. The ISSRM *assets* are modelled using Secure Tropos constructs *Goal, Softgoal, Actor, Plan* and *Resource*. *Goal* is defined a desired state that an actor is determined to achieve (e.g., Data Employed in Fig. 2). *Softgoal* is a desired state that an actor is determined to achieve yet there is no clear determination of how this state is to be achieved (e.g., Confidentiality & Integrity in Fig. 2). *Actor* is an entity that is part of a system and is driven by certain goals and intentions (e.g., Server and Input Interface in Fig. 2). *Plan* is a course of action followed by an actor in order to achieve and satisfy a goal (e.g., Submit data in Fig. 2). The relationships between the assets are modelled using the constructs of *contribution*, *means-ends*, and *decomposition*. The ISSRM *security criterion* is represented by combining a *Softgoal* with *Security constraint*(*s*) (e.g., Confidentiality & Integrity and Maintain the integrity & confidentiality of the submitted data in Fig. 2). The ISSRM *constraints of* relationship can be modelled both explicitly by the *Restrict* link (see Fig. 2) and implicitly as security constraint placed on the security dependency link and restricting use of *dependum* (e.g., see connection between Server and Input Interface in Fig. 2).

Risk-related concepts. To distinguish risk related concepts darker colours are introduced to Secure Tropos constructs. The ISSRM *threat agent* is represented as actor (e.g., Attacker in Fig. 5). The ISSRM *attack method* – as a *plan* and the ISSRM *threat* as a combination of *goal* and *plan* (e.g., submitted data obtained and Intercept transmission in Fig. 5). The ISSRM *vulnerability* is not represented, how it is indicated through *vulnerability point* (see black circle in Fig. 5).

Risk treatment-related concepts. The ISSRM *security requirements* are modelled by combining constructs of *Goal*, *Softgoal*, *Plan*, and *Security constraint* (e.g., (S)

Perform Cryptographic procedures in Fig. 6). The ISSRM *mitigates* relationship is used to indicate a connection where a construct or group of constructs mitigate a certain security risk.

3 Security Patterns

"*A security pattern describes a particular recurring security problem that arises in specific contexts, and presents a well-proven generic solution for it*" [12]. Following this definition and the principles of the security risk management (see Section 2.1), five security risks-oriented patterns are introduced in [2]. In this section we briefly recall these patterns and illustrate how RAST could be used to represent them.

3.1 Security Risk-oriented Patterns

SRP1 describes how to *secure the transmission of confidential data between business entities*. This pattern involves an attacker who intercepts the transmission between the input interface and the server, then obstructs and modifies the data. The attack is facilitated due to the transmission medium not being encrypted and data being stored in a plain text. The risk event leads to the loss of the confidentiality of the data and loss of the integrity of the data. The risk is mitigating by introducing cryptographic and checksum countermeasures.

SRP2 enables *validation of data submitted to a business activity, by predicting the need for a mechanism that scans and detects malicious data before the data is forwarded to this business activity*. This pattern counters an attacker that has information regarding the systems inner functionalities. The malicious agent attacks by submitting through the input interface a malicious script that exploits the fact that incoming data are not filtered. The attack leads at the loss of confidentiality and the integrity of the business activity that is forwarded to.

SRP3 ensures the *availability of a service in a Denial of Service* (DoS) event. The attacker sends an exponentially growing number of simultaneous requests to the system, resulting in the system crashing due to its ability to only serve a certain number of simultaneous clients. The attack leads to the loss of the service availability.

SRP4 focuses on *securing confidential information, from being accessed by unauthorised devices or people*. An attacker gains access to sensitive business data through a commonly used retrieval interface. Due to the interface not having an access control mechanism, the attacker is able to retrieve the data. The attack negates confidentiality of the business data.

SRP5 specifies how to *secure data stored into a business data store* against internal attacks. The attack occurs due to the data being stored in a plain format, and, thus, leads to the loss of the confidentiality of the stored data and the perpetual damage of the files residing in the same instance as malicious script.

3.2 Security Risk-oriented Patterns Expressed in Secure Tropos

In this section we demonstrate how RAST could be applied to represent SRPs; more specifically we will represent SRP1. For instance, in Fig. 2 we define Submitted data as the ISSRM *business asset*. Both Server and Input Interface should collaborate in the way to achieve Confidentiality and Integrity of the submitted data. This *security criterion* is clarified by security constraint Maintain the integrity & confidentiality of the submitted data. This constraint restricts the goal of Data employed at the Server side and Data submitted at the Input interface side. Submit Data plan is the dependum between the two actors, and two constrains indicate that this double constrained dependency should be fulfilled by Server and Input Interface's activities.

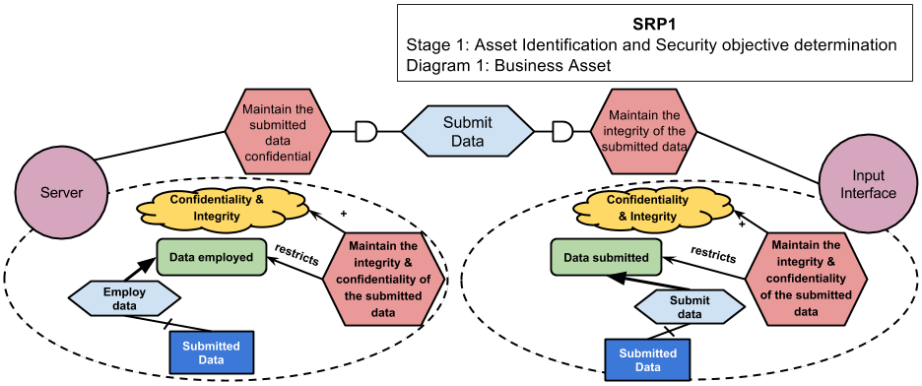

Fig 2. SRP1: Assets and Security Criteria

Fig. 3 introduces the Transmission Medium actor, which is used to transfer data from Input Interface to Server. This actor is part of the considered system (i.e., *IS asset*), thus, it is used to support the transfer of the *business asset* (i.e., Submitted data).

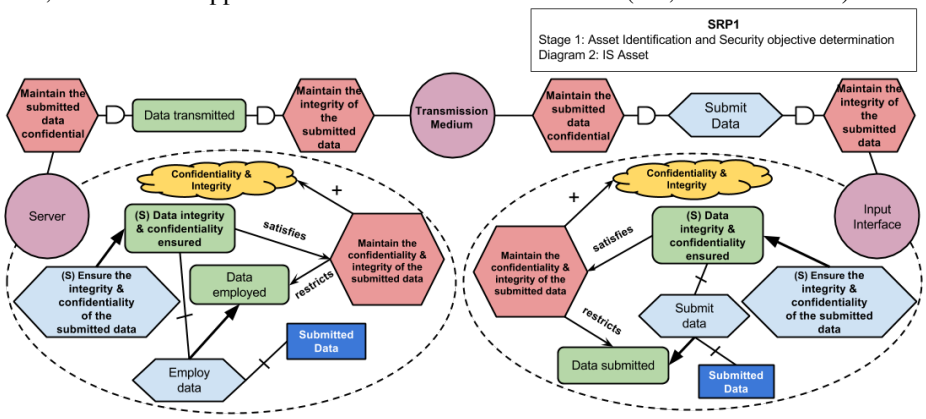

Fig 3. SRP1: Context Pre-processing

In Fig. 4 we identify a security *even* defined as Man in the middle attack that *impacts* the security criterion Confidentiality & integrity. In Fig. 5 this event is expanded

showing how Attacker can achieve his goal Submitted data obtained by executing the *attack method* Intercept transmission.

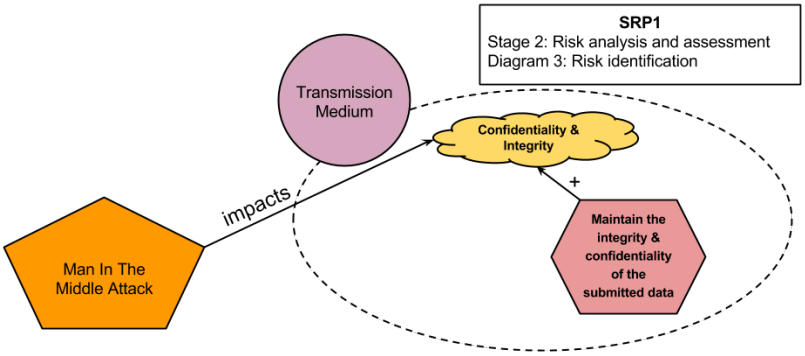

Fig 4. SRP1: Security Risk Identification

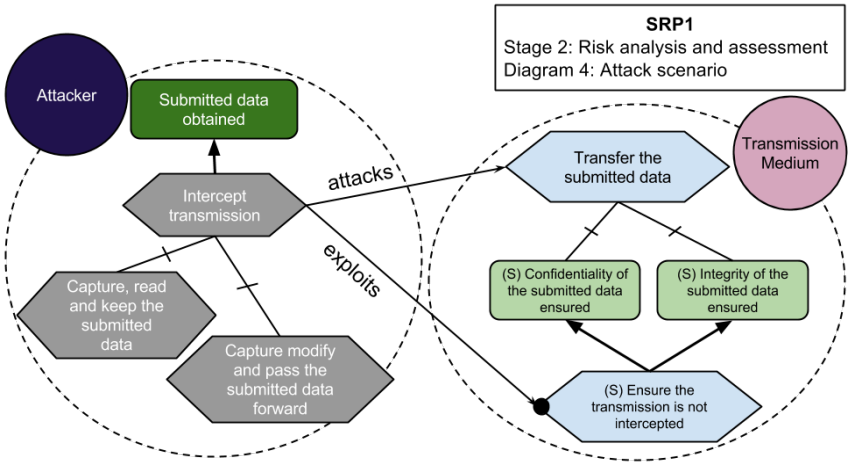

Fig 5. SRP1: Security Threat

To mitigate the risk in Fig. 6 we present the implementation of *risk reduction* decision. Hence there, the secure plan of the Ensure the integrity & confidentiality of the submitted data (see Fig. 3) is changed with Perform cryptographic procedures and Perform checksum procedures. The replacements are performed in the according actor of the model. Risk mitigation is indicated using the Mitigates relationships.

Although RAST contains some limitations with respect to the ISSRM domain (as indicated in [7][8]), the language allows represent the SRP description. In our example, the recurring security problem is illustrated in Fig. 4 and 5, the context in Fig. 2 and 3. Finally we present the solution in Fig. 6. We will illustrate how graphical SRP representations can be used to derive security requirements from the Secure Tropos models in Section 4.

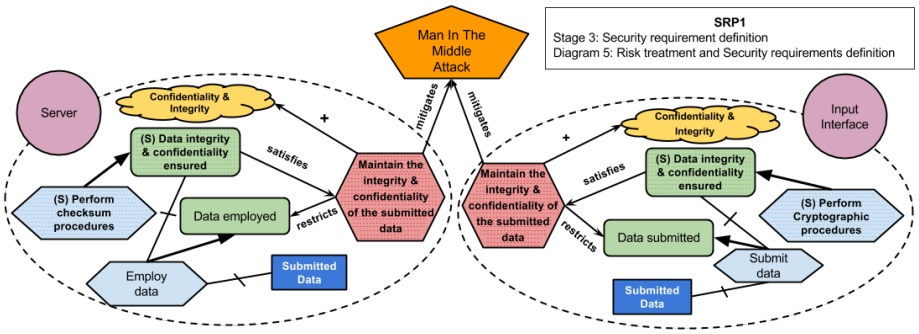

Fig 6. SRP1: Security Requirements Definition

4 Deriving Security Requirements using SRP's

4.1. Collaboration Between System and Security Analysts

Application of SRPs to Tropos model could stimulate collaboration between two roles (as illustrated in Fig. 7) – *system analyst*, who is responsible for system development, for example, using *i**/Tropos method, and *security analyst*, who is responsible for security solutions and could potentially apply SRPs to achieve her goals. In some cases both roles could be played by the same person. For instance, after creating system model using *i**/Tropos method, system analyst could potentially request security analyst to determine security requirements. After analysing the system model, security analyst selects and applies the relevant SRPs. The SRP application includes (1) SRP occurrence identification and asset alignment, (2) vulnerable asset identification and secure goal introduction, and (3) security requirements introduction. After this iteration, security analyst could potentially consider whether other SRPs should be applied. If not the system model with introduced security requirements is returned back to system analyst.

Next system analyst should potentially decide which security requirements could be implemented to the targeted system. In other words, system analyst needs to perform trade-off analysis to understand the cost-value benefits of the security solution. In case of necessity, system analyst could potentially request security analyst for justification of the proposed security requirements. In the latter case, the instantiated SRP's security threat models (e.g., see Fig. 5) could be used to (4) provided security requirements rationale.

4.2. Security Requirements Derivation

Now we will illustrate how the SRPs expressed in RAST could be used to derive security requirements from the Secure Tropos models. The model [3] used to demonstrate the derivation process is presented in Fig. 8. We will use the SRP1 (illustrated in Section 3.2). However other SRPs can be following the same steps.

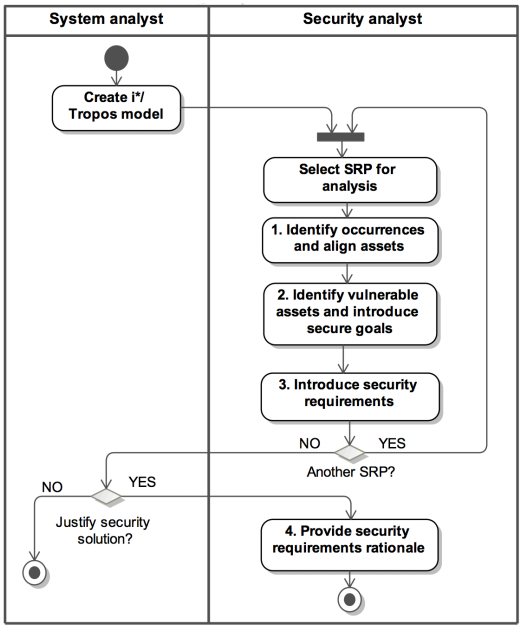

Fig 7. SRP1: Collaboration between System and Security Analysts

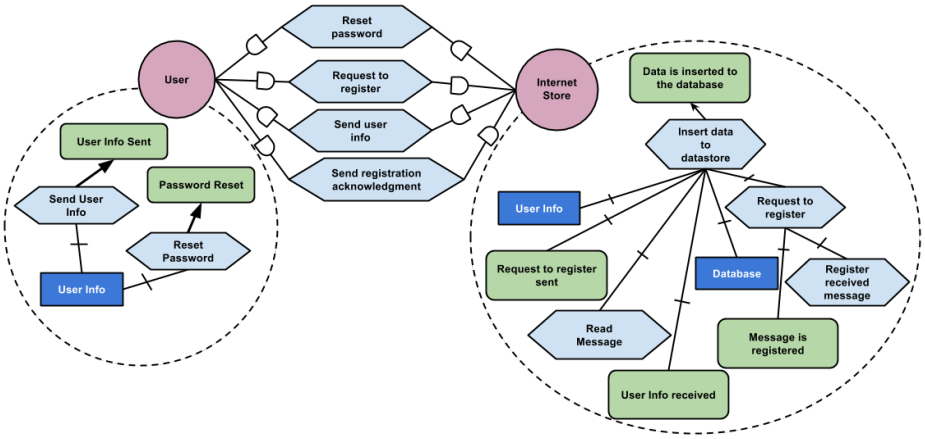

Fig 8. Internet Store Registration; adapted from [3]

1. Occurrence identification and asset alignment. This step includes identification of the pattern occurrences in the analysed model. This is a manual activity due to the complexity of the Secure Tropos models. In Fig. 8 we aim to apply SRP1. The contextual pattern description helps to observe the following occurrences:

- The Internet Store actor aligns to the SRP1 Server actor, thus, giving the similar interactions with the other actors;

- The User (Interface) aligns to the SRP1 Input Interface due to the connection to the Internet Store/Server. Given that a 1:1 occurrence not existing between SRP1 and the scenario under investigation, we assume that the User fulfils the Send Use Info plan by using an input interface provided by the Internet Store. This is why we recall the User actor as User Interface.

Following the pattern description a Transmission Medium is introduces as the intermediate actor to Send user info. This assumption is done to support the communication between the User and the Internet Store.

Next step is to consider the dependency relationship and potentially equipped it with the security constraints in order to highlight the *security objectives*. The extracted occurrence of the pattern is illustrated in Fig. 9.

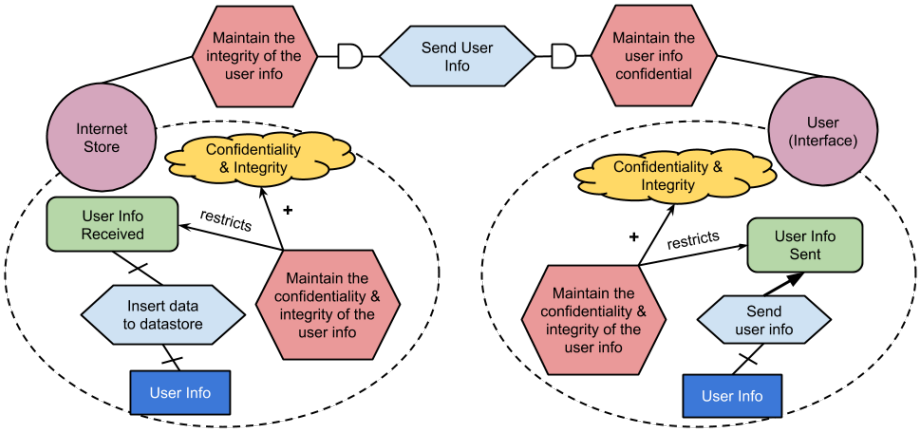

Fig 9. Asset and Security Criteria Analysis

2. Vulnerable asset identification and secure goal introduction. In this step vulnerable assets are identified and security criteria and constraints are explicitly introduced. The process follows the RAST methodology of separately illustrating business and IS Assets as illustrated in Fig. 9 and 10. Additionally in this instance we introduce the secure goals and secure plans suggested by the pattern. The secure goals and plans are introduced to their aligned goals.

3. Security requirements introduction. Following the SRP1 risk treatment and security requirements definition, it becomes possible to introduce secure goals and plans as illustrated in Fig. 11. The previously defined model now is also equipped with *security requirements* such as Perform checksum procedures and Perform Cryptographic procedures. As illustrated in Fig. 11 both suggestions *mitigate* the security event (i.e., Man in the Middle Attack).

4. Security requirement rationale. In some cases this step could be considered as optional, but it becomes important, once one needs to understand the rationale and trade-off of the newly introduced security requirements. Following the pattern attack scenario, in this step one defines how the security threat could be carried on the targeted system. More specifically in our case, Fig. 12 illustrates how Attacker (i.e., *threat agent*) could obtain the user info by intercepting transmission.

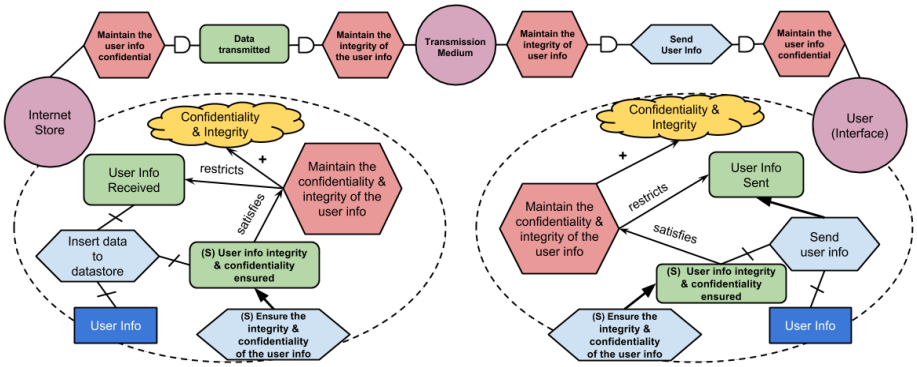

Fig 10. Model Pre-processing

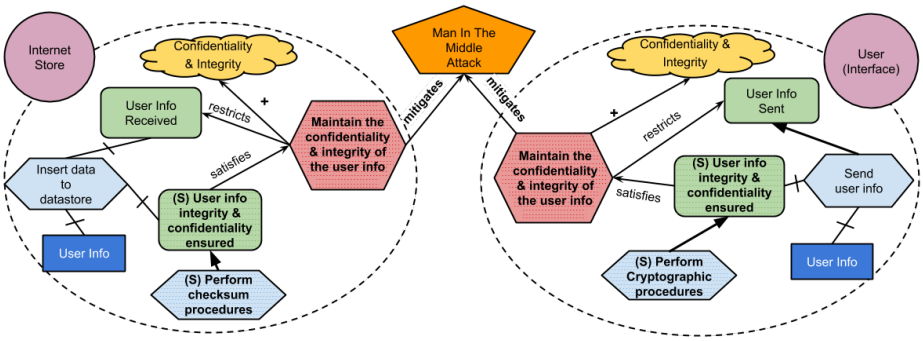

Fig 11. Security Risk Identification

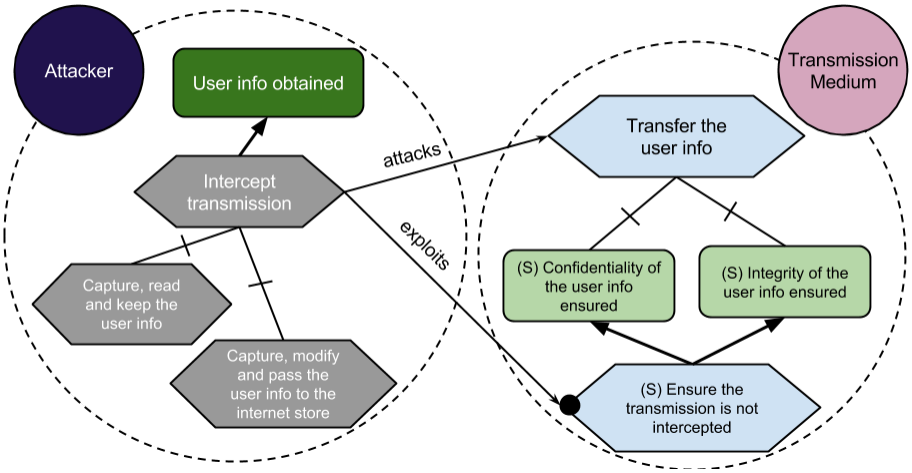

Fig 12. Security Threat

5 Validation

To validate the SRPs expressed in RAST and their application process contribution we have conducted an observatory study. Its **objectives** were (*i*) to understand *correctness* and (*ii*) *usability* the SRP application, (*ii*) to compare *understandability* of the SRP application of the participants with the ISSRM background against participants without ISSRM background.

5.1 Observatory Study Design

Participants. We have invited six individuals with the software engineering background. Three participants (i.e., *group A*) had the IS-security background as they were working in the field of enterprise security and had prior knowledge of the ISSRM concepts. Other three participants (i.e., *group B*) had no information systems security background, but nevertheless they were practitioners working in the software engineering companies.

Design. Firstly, the participants were given the introductory lecture. Secondly, they participants were given a Secure Tropos model and were asked to derive the security requirements using the SRPs. Finally, the participants filled the questionnaire on the usability of the SRPs and their application process. Each participant took approximately 3 hours to complete the process.

Treatment. The lecture included an introduction of the ISSRM domain model and security risk management process (see Section 2.1), RAST (see Section 2.2), SRPs (see Section 3) and their application process (see Section 4). The lecture was concluded with an SRP application demo.

SRP application task focussed on the pattern application process (see Section 4). Participants were given a model described in Section 3 and were requested to identify SRP occurrences as well as derive security requirements. None of the participants applied all the SRPs (see Table 1), since we limited their participation to three hours.

Questionnaire. Once the SRP application was completed, participants filled the questionnaire on the usability of the RAST process, SRPs and SRP application process. Specifically in included questions on easiness, satisfaction, and understandability of used artefacts.

5.2 Threats to Validity

The following threats to validity should be taken into account:

- The number of participants is rather small (only six participants) thus the sample may not be accurate. The results might differ if we were able to attract more participants. However they all were practitioners working in the field of software security engineering.
- Given treatment could influence the received results. When applying SRPs participants had some questions, and we provided then with the answers. However otherwise they would have difficulty to complete the given task.

Each participant conducted his task individually. If participants had opportunity to discuss and to learn from each the result potentially would be different.

- Each participant applied different SRPs. Ideally all of the participants would have to complete all SRPs for better result. However, we were limited by the time constraints (three hours per participant).
- Participants had a varying level of prior knowledge of ISSRM domain and security risk management. Having participants with the same knowledge could potentially deliver more reliable result. We tried to mitigate this threat by provided introductory lecture.
- The majority of the participants implemented the models using an online drawing tool. Implementing the models by hand or other modelling tool method could impact the modelling outcome.
- The participants were not told that they were expected to perform in a certain way or that a specific result was expected from them. Stating expectations upfront would impact the overall performance of the participants. The performance could be enhanced in case the participant would want to perform according the expectations. Or the participant could suffer from a type of performance anxiety and his result would be negatively affected.
- Participant had prior acquaintance with the conductor of the observatory study and the first author of this paper. If no prior acquaintance would occur participants could not ask the same questions or perform in the same manner they would perform to another individual. However this acquaintance was the way to involve participants in the evaluation.
- Some SRPs were easier identified in the model comparatively to other SRPs. If all the SRPs would be identical in terms of identification ease, the result could be different. Making an SRP easier or harder to identify results in the pattern application process becoming automatically easier of harder to perform.

All the above threats had a certain effect to the overall results. We assume that in case of a more extensive study with a greater number of participants and different design, different outcomes might be received.

5.3 Observatory Study Results

Correctness of the participant models. Correctness of each SRP application is defined through the number of errors identified in the resulting model (i.e., the lower number indicates better model correctness). Errors are divided to two categories *phrasing* and *modelling* errors. Phrasing errors describe any error in regards to the phrasing of any of the constructs (e.g. labels of goals, plans, and etc.). Modelling errors describe errors performed in the modelling of each concept. Modelling errors include using wrong constructs when linking assets, risk components, security countermeasures and similar. Additionally, modelling mistakes also include incorrect colouring of the constructs (as the colour here brings the semantic difference between security risk concepts). Both types of errors are discovered by comparing the participants' models with the models prepared by the first author of this paper.

Table 1 presents the result of the model correctness. It was observed that *Participant 2* has made the least amount of errors compared to the other participants.

It is also important is to point out that the majority of errors done by *Participant 1* were rather minor in comparison to phrasing errors of the other participants. In general the majority of the errors are phrasing errors. This could be explained by the fact that the modelling language does not provide explicit guidelines on how to name the constructs during modelling.

Table 1. Participant Pattern Application Errors

Participants	Applied SRP	Phrasing Errors	Modelling Errors	Total
Participant 1	SRP2	20*	0	20*
	SRP4	-*	-*	-*
Participant 2	SRP5	0	4	4
Participant 3	SRP1	11	0	11
	SRP3	16	0	16
Participant 4	SRP4	11	0	11
Participant 5	SRP1	26	13	39
Participant 6	SRP2	10	0	10

* Not eligible for error counting due to the participant not using an existing construct but assumed that the system includes the functionality.

Understandability. As mentioned in the design description, the participants were divided to two groups – *group A* and *group B* – based on their previous experience with the security engineering. The results show that participants of *group A* were able to apply and comprehend the used patterns as well as the pattern application process. Participant of *group B* were able *moderately* to apply and to comprehend SRPs. *Group A* correctly executed all the pattern application steps. Nonetheless mistakes were made in phrasing and resource decomposition (as discussed above). But they performed all the tasks in a rather reasonable time and were confident in their results.

Group B completed the SRP application process with a *moderate* correctness. Similar to *group A*, *group B* also made mistakes in phrasing and modelling. Furthermore, noticeable difference in the results was the level of confidence in the results of the application process. Participants of *group B* were notably less confident than the participants of *group A* in their results. As conclusion we observed that the information systems security experience had some impact and helped better contributed to the understandability of the SRPs application.

Discussion and Concluding Remarks. We draw out the concluding remarks based on our observations and the participant responses marked in the questionnaire. All participants completed the application of at least one pattern. Mistakes were observed in the phrasing and modelling of various assets of the models. In comparison less mistakes were made in modelling rather than phrasing. All the participants understood the proposed SRP representations. The pattern application process was according to the majority of the participants moderately easy to be applied. RAST affects the overall process in a moderate level.

The fact that both groups were able to complete the tasks assigned, demonstrated that the process is useable as a starting point to derive security requirements in a goal-oriented environment. The easiest part in the application process according to the majority of the participants was the pattern identification and asset alignment. The

hardest step to be applied by the majority of the participants was security requirement introduction and extracted model re-integration.

Having background knowledge in IS security affects the process during the first applications and speeds up moderately the security requirement derivation process. Prior knowledge of an agent-oriented language in combination ISSRM affects rather positively the outcome of modelling. Participant that had no prior knowledge were less confident about their results. The following lessons are learnt:

- Application of the SRPs helps to construct rather correct security models and derive appropriate security requirements. The major modelling mistakes are made due to the lack of guidance from the modelling language application.
- The SRP application guidelines are rather understandable and moderately usable by their users. However, a priori experience in security engineering helps to see the method purpose. Potentially some security engineering training could help to improve method application.

6 Related Work

There are few studies where the secure *i** framework [13] or Secure Tropos is used to capture security and privacy requirements through security patterns. Some extensions are proposed to Secure Tropos to be suited for the security pattern description language [10]. Elsewhere in [5], legal requirements are incorporated to security and privacy patterns expressed using another extension of Tropos methodology towards security. Here authors concentrate on access control, need-to-know, outsourcing, and non-repudiation patterns. In addition to these contributions, in this paper we use the Secure Tropos approach [11] to represent security risk-oriented patterns.

In [1] Ahmed has presented a method for security requirements elicitation from business processes (SREBP). This method enables security requirements derivation from BPMN models, namely value chains and business process models. The method involves collaboration between the business analyst and security analyst. In the current study we develop a method to derive security requirements from the RAST models. Hence the collaboration is defined between the security analyst and system analyst, since we consider the *late requirements* stage modelled in Secure Tropos.

7 Summary and Future Work

In this paper we analyse how to integrate security risk-oriented patterns into the goal-oriented IS development. We have developed a threefold procedure. Firstly, it is important to define and describe the SRPs in modelling language used for IS development, in our case RAST. Secondly this description is used to identify the pattern occurrences in the targeted IS model. Typically this step requires some model pre-processing. Finally, the security requirements are derived and introduced following the SRP description. To support this procedure we have presented a pattern presentation structure as well as their application process. The proposal is validated in

the observatory study, which illustrates the SRP usability. Finally it was demonstrated that the proposed SRP's could potentially be the starting point for security requirements derivation and security trade-off analysis.

The future work includes expansion of the SRP list with new patterns and their representation using RAST. Also it is important to define guidelines for systematic security requirements prioritisation and their implementation to security controls. Last but not least, the software tools to support the representation of the SRPs ain RAST and their application process could potentially help to decrease the application effort.

Acknowledgement

This research is supported by the Estonian Research Council.

References

1. Ahmed N.: Deriving Security Requirements from Business Process Models, *PhD thesis*, *University of Tartu*, 2015
2. Ahmed N., Matulevičius R.: Securing Business Processes using Security Risk-oriented Patterns, *Computer Standards & Interfaces*, 36 (4), 2014, pp 723-733
3. Altuhhova O. 2013 An Extension of Business Process Model and Notation for Security Risk Management, *McS thesis, University of Tartu*, 2013
4. Bresciani P., Perini A., Giorgini P., Giunchiglia F., Mylopoulos J.: Tropos: An Agent-Oriented Software Development Methodology, *Autonomous Agents and Multi-Agent Systems*, 8, 2004, pp 203-236
5. Compagna L., El Khoury P., Krausov A., Massacci F., Zannone N.: How to Integrate Legal Requirements into A Requirements Engineering Methodology for the Development of Security and Privacy Patterns, *Artificial Intelligence and Law*, 17 (1), 2009, pp 1-30
6. Dubois E., Heymans P., Mayer N., Matulevičius R.: A Systematic Approach to Define the Domain of Information System Security Risk Management, *Intentional Perspectives on Information Systems Engineering*, *Springer Berlin Heidelberg*, 2010, pp 289-306
7. Matulevičius R., Mayer N., Mouratidis H., Dubois E., Heymans P., Genon N.: Adapting Secure Tropos for Security Risk Management in the Early Phases of Information Systems Proceedings of CAiSE 2008, Springer Berlin Heidelberg, 2008, pp 541-555
8. Matulevičius R., Mouratidis H., Mayer N., Dubois E., Heymans, P.: Syntactic and Semantic Extensions to Secure Tropos to Support Security Risk Management, *J. UCS*, 18, 2012, 816-844
9. Mayer N.: Model-Based Management of Information System Security Risk, *PhD thesis*, *University of Namur*, 2009.
10. Mouratidis H., Giorgini P., Schumacher M., Manson M.: Security Patterns for Agent Systems, Proceedings of EuroPLop 2003.
11. Mouratidis H., Giorgini P.: Secure Tropos: A Security-oriented Extension of the Tropos Methodology. *International Journal of Software Engineering and Knowledge Engineering* (*IJSEKE*), 17(2), 2007, pp 285–309
12. Schumacher M., Fernandez-Buglioni E., Hybertson D., Buschmann F., Sommerlad P.: Security Patterns, Integrating Security and Systems Engineering, *Wiley* 2006
13. Yu E.: Towards Modeling and Reasoning Support for Early-phase Requirements Engineering. *Proceedings of the 3rd IEEE Int. symposium on Requirements Engineering*, *IEEE Computer Society Press*, 1997, pp. 226–235.