predict
prioritise
prevent

TRESPASS

Technology-supported Risk Estimation by Predictive Assessment of Socio-technical Security

Deliverable D1.1.2

Final Specifications and Requirements for Socio-Technical Security Models

Project:	TRE_SPASS
Project Number:	ICT-318003
Deliverable:	D1.1.2
Title:	Final Specifications and Requirements for Socio-Technical Security Models
Version:	1.0
Confidentiality:	Public
Editor:	C.W. Probst
Cont. Authors:	C.W. Probst, J.W. Bullee, L. Montoya, M.G. Ivanova, O. Gadyatskaya, R.R. Hansen
Date:	2015-10-30

Part of the Seventh Framework Programme
Funded by the EC-DG CONNECT

Members of the TRESPASS Consortium

1.	University of Twente	UT	The Netherlands
2.	Technical University of Denmark	DTU	Denmark
3.	Cybernetica	CYB	Estonia
4.	GMV Portugal	GMVP	Portugal
5.	GMV Spain	GMVS	Spain
6.	Royal Holloway University of London	RHUL	United Kingdom
7.	itrust consulting	ITR	Luxembourg
8.	Goethe University Frankfurt	GUF	Germany
9.	IBM Research	IBM	Switzerland
10.	Delft University of Technology	TUD	The Netherlands
11.	Hamburg University of Technology	TUHH	Germany
12.	University of Luxembourg	UL	Luxembourg
13.	Aalborg University	AAU	Denmark
14.	Consult Hyperion	CHYP	United Kingdom
15.	BizzDesign	BD	The Netherlands
16.	Deloitte	DELO	The Netherlands
17.	Lust	LUST	The Netherlands

Document History

Authors		
Partner	Name	Chapters
UT	Lorena Montoya	2
UT	Jan-Willem Bullee	2
DTU	Marieta Georgieva Ivanova	2
DTU	Christian W. Probst	All
UL	Olga Gadyatskaya	2
AAU	René Rydhof Hansen	3
TUHH	Dieter Gollmann	Quality Assurance

Quality assurance		
Role	Name	Date
Editor	Christian W. Probst	2015-09-30
Reviewer	Lars Wolos	2015-09-15
Reviewer	Roel J. Wieringa	2015-09-15
Task leader	Christian W. Probst	2015-10-30
WP leader	Christian W. Probst	2015-10-30
Coordinator	Pieter Hartel	2015-10-30

Circulation	
Recipient	Date of submission
Project Partners	2015-10-30
European Commission	2015-10-30

Acknowledgement: *The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318003 (TRESPASS). This publication reflects only the authors' views and the Union is not liable for any use that may be made of the information contained herein.*

Contents

List of Figures

List of Tables

Management Summary

Key takeaways:

- We have identified a wide spectrum of required properties that socio-technical security models should be able to represent.
- Socio-technical models fulfilling these requirements enable formal analyses and visualisations that contribute to attack navigators.
- The TRESPASS model provides a unique combination of properties, that extend or complement existing models, making the TRESPASS model attractive for researchers and practitioners.

The primary goal of the TRESPASS project is to develop tools that facilitate assessment and management of IT security-related risks in organisations, spanning both technological and sociological issues. To support this goal, work package WP1 provides a socio-technical security model that forms the basis of the TRESPASS tools, especially the TRESPASS attack navigator. The model developed by WP1 must support robust, yet flexible, modelling of organisations as socio-technical entities containing such diverse aspects as IT infrastructure, organisational structure, and physical structures as well as human factors.

This document presents final findings of the TRESPASS project for requirements for socio-technical models, based on a structured literature review of existing models, and the possibilities and the limitations of these models.

1. Introduction

This document provides the final requirements for work package WP1 ("Socio-technical security model specification") of the TRESPASS project. Appendix A provides the context for this deliverable in the TRESPASS project. It describes the overall summary of the project and the TRESPASS workflow.

The objective of work package WP1 is to develop models that capture the essentials of an organisation and its structure on three core levels: the physical, digital, and social domains. The model contains entities, attributes, and relations that are relevant for analysing the organisation's security.

The socio-technical security models are at the heart of the technical part of TRESPASS, and constitute the interface between the organisation being modelled and the TRESPASS processes and tools developed in other work packages, such as the data collected in WP2, the analysis tools of WP3, and the visualisation tools of WP4. For the integration into risk assessment frameworks (WP5), the models developed in WP1 are the entry point into the TREsPASS process (The TRESPASS Project, D5.1.2, 2015).

Models of organisational infrastructures have been used before, *e.g.*, to identify insider threats in organisations and to compute attacks on organisations. The challenge in this work package is to integrate the modelling of policies and social dimensions, and again to do so in a way that supports implementation, analysis, and visualisation.

Socio-technical security models are therefore a key enabling factor for the TRESPASS process; they must be easy to develop and to visualise, to enable practitioners to interface with the TRESPASS tools, and they must be detailed enough to support the analysis tools to predict attacks, prioritise attacks, and suggest preventive measures. Especially the formal treatment of models is important, and requires structured, well-founded models.

An important aspect of the models is modularity, not only to support modular model development and maintenance (WP5), but also to support compositional analysis of the models being developed (WP3). The TREsPASS socio-technical security models will also be modular in the sense that different features can be added on demand; features such as detective components, for example, are optional and only added when needed for modelling the organisation or to model a new threat scenario (The TRESPASS Project, D1.3.2, 2015; The TRESPASS Project, D1.3.3, 2015).

1.1. Goals

The goals of this work package are the development of a model and methodology to

- Support the access to the social and technical data collected by WP2;
- Support the analyses and visualisations developed in WP3 and WP4;
- Support the TRESPASS tools and processes developed in WP5 and WP6; and
- Support the modelling of case studies in WP7.

This documents presents the requirements for the work package based on these goals. The report builds upon a previous deliverable D1.1.1 “Initial requirements for socio-technical security models” (The TRESPASS Project, D1.1.1, 2013), and contains some of the earlier material. Section 1.4 discusses the relation to this material.

1.2. Foreground and background

All the findings presented in this deliverable are foreground.

1.3. Document structure

The rest of this document is structured as follows. After a discussion of the relation of the present deliverable with its predecessor D1.1.1 (The TRESPASS Project, D1.1.1, 2013), we present an overview of the results of the literature review performed in preparing this deliverable.This is followed by an overview of the identified requirements in Chapter 3.

Before concluding, we present an overview of the requirements identified to and from WP1 in TRESPASS in Chapter B and conclude in Chapter 4.

Section 1.4 and Chapter B are specific to TRESPASS internals and will not be part of the public version of this deliverable. They are included here for the M36 review.

1.4. Relation to Deliverable D1.1.1 (not public)

Since the submission of Deliverable D1.1.1 (The TRESPASS Project, D1.1.1, 2013), we have concluded our structured literature review, which in excerpts is presented in Chapter 2, superseding Chapter 2 from the first iteration of this deliverable. Chapters 3 and 4 from D1.1.1 were included there to bootstrap the development of the TRESPASS tools. Attack trees where consequently covered in WP3, and ArchiMate has become a separate tool to develop TRESPASS models. As such, neither is relevant for this deliverable.

The requirements identified in Chapter 5 of D1.1.1 are superseded by the requirements described in Chapters 3 and B.

2. Literature Review

This section presents the result of a structured literature review in the area of security modelling, to identify the needs for the TRE$_S$PASS socio-technical security model. We are therefore especially interested in the possibilities to incorporate social, technical and physical infrastructures within models. Consequently we address the questions: **What features do current security models have?** and **Which features are needed for the TRE$_S$PASS socio-technical security model?**

2.1. Method

We have performed a systematic literature review (Kitchenham, 2004) to identify relevant existing literature on socio-technical security modelling languages, and to summarize the current state of this topic and highlight the challenges. A review protocol describing each step of the review, including eligibility criteria, was developed before beginning the search for literature and the data extraction.

We considered articles covering aspects of security modelling of socio-technical systems and keywords from the results of an initial search (Pieters, 2011; Sommestad, Ekstedt, & Holm, 2012; Dragovic & Crowcroft, 2004, 2005; Dragovic, 2006; Hunker & Probst, 2011; Probst & Hansen, 2008; Probst & Hansen, 2009; Sameer, 2011; De Nicola, Ferrari, & Pugliese, 1998; Gorla & Pugliese, 2003; Bettini, Loreti, & Pugliese, 2002; Probst, Hansen, & Nielson, 2007; Mathew et al., 2005; Mathew, Upadhyaya, Ha, & Ngo, 2008; Franqueira, Lopes, & van Eck, 2009; Dimkov, Pieters, & Hartel, 2010; Dimkov, 2012; Scott, Beresford, & Mycroft, 2003; Scott, 2004). These articles covered nine models, containing together 134 keywords, from which the most relevant were selected and combined, to increase relevance of the query results. The decisions on articles, keywords, and their combination were made after intensive discussions with experts inside and outside of TRE$_S$PASS. In the end we used the following queries for the review:

- attack $\wedge$ model $\wedge$ scenario $\wedge$ socio-technical
- cyber $\wedge$ model $\wedge$ socio-technical $\wedge$ vulnerability
- cyber-attack $\wedge$ insider $\wedge$ model
- cyber attack $\wedge$ model $\wedge$ scenario $\wedge$ vulnerability
- cyber-attack $\wedge$ model $\wedge$ socio-technical $\wedge$ vulnerability
- cyber-attack $\wedge$ security modelling

- cyber-attack $\wedge$ security modelling $\wedge$ vulnerability analysis
- security modelling $\wedge$ socio-technical

The search was applied to the SCOPUS database, which also covers publications from Cambridge University Press, Elsevier, Springer, Wiley-Blackwell and the IEEE. The automated search was carried out in 2013 and 2014, and the results of the queries were filtered based on a first read of the articles and an assessment of inclusion criteria. We also scanned the reference lists of the papers in order to identify relevant other sources. In addition we have interviewed domain experts within TRE$_S$PASS and outside of the project, who have suggested examining some additional studies.

An overview of the results is given in Table 2.1 in the end of this section.

2.2. Representation

The approaches considered in this review represent models either graphical or textual. The graphical models can be divided in tree and graph structures, diagrams, and map overviews.

The tree structured models include the work by Dragovic et al. (Dragovic & Crowcroft, 2005) and Scott *et al.* (Scott, 2004), where trees represent the world, and the MsAMS framework (Franqueira et al., 2009), where trees represent network topologies. Attack trees model all steps that need to be taken to achieve the main goal of the attack. The root node of an attack tree depicts the goal of the attacker, whereas the children of a node in the tree are refinements of the node's goal into sub-goals. The leaves of the tree represent the basic actions to be executed by the attacker (Karpati, Opdahl, & Sindre, 2011; Pieters, Dimkov, & Pavlovic, 2013; Ten, Manimaran, & Liu, 2010; Vintr, Valis, & Malach, 2012). Alternatively, attack patterns describe generic approaches used by attackers (Karpati et al., 2011). Finally, fault trees are used to represent failure information about systems (Vintr et al., 2012). Boolean logic Driven Markov Processes are an extension of Fault Tree with Markov processes (Kriaa, Bouissou, & Pietre-Cambacedes, 2012).

Graph structures are used to construct Capability Acquisition Graphs, presented by a tuple containing of Vertices, Edges and System properties (Mathew et al., 2008). In the CySeMol model, the graph structure is used in a Reachability graph to link steps in an attack (Sommestad, Ekstedt, & Johnson, 2010; Sommestad et al., 2012). Hyper graphs are used in the ANKH model present membership of a group (Pieters, 2011) and in the MsAMS language to define broadcast communications (Franqueira et al., 2009). Portunes (Dimkov et al., 2010; Dimkov, 2012) and ExASym (Probst et al., 2007; Probst & Hansen, 2008; Probst & Hansen, 2009) use directed graphs connect data or places of interest in the model. A totally ordered graph is used as simultaneous attacks graph to represent coordinated attacks (Samarji, Cuppens, Cuppens-Boulahia, Kanoun, & Dubus, 2013). Directed Acyclic Graphs (DAG) are used to model Attack Graphs, containing paths an attacker could use to achieve his goal (Xie, Chen, Wang, Chen, & Hu, 2009; Sarkar,

Kohler, Riddle, Ludäscher, & Bishop, 2014), and directed bipartite graphs are used to visualize Petri Nets, and are well suited for modelling distributed systems and concurrent behaviour (Sanders & Freire, 1993).

Diagrams also come in different flavours. Attack sequence diagrams describe an intrusion from the intruder's point of view as a sequence of ordered steps. Each step describes an attacking activity to be used in that steps (Karpati et al., 2011). Data flow diagrams are used to create threat models (Shostack, 2008). Misuse case maps focus on vulnerabilities, threats and intrusions from an architectural point of view, and are an extension of Use Cases , with security elements (Karpati et al., 2011). The related Misuse sequence diagram graphically show an intrusion sequence as a combination of misuse cases and UML sequence diagrams, helping to analyze complex intrusion scenarios (Karpati et al., 2011).

Finally, ExASyM uses a building blue print as basis for the model (Probst et al., 2007; Probst & Hansen, 2008; Probst & Hansen, 2009; Sameer, 2011). Misuse case maps present security issues from an architectural perspective, and are the combined perspective of misuse cases and use case maps. They combine perspectives from misuse cases and use case maps, providing a combined overview of a software system's architecture and its behaviour by drawing usage scenarios paths (aka use cases) (Karpati et al., 2011).

Textual notations for models are often used for elements in models, for example processes. The algebraic process calculus is an overview description of the processes and communication of a system. This allows formal reasoning about behaviour and the system. One example of an algebraic process calculus is KLAIM, the Kernel Language for Agents Interaction and Mobility (De Nicola et al., 1998), which is the basis for both ExASyM (Probst et al., 2007; Probst & Hansen, 2008; Probst & Hansen, 2009; Sameer, 2011) and Portunes (Dimkov et al., 2010; Dimkov, 2012). The situational calculus is used to model and analyse coordinated attacks (Samarji et al., 2013), and temporal logic allows to reason about time aspects (Shahriari, Makarem, Sirjani, Jalili, & Movaghar, 2010).

2.2.1. Discussion

There are two main categories found to represent model: graphical and textual. The graphical representation can be divided in trees, graphs, diagrams, and maps, with the former two having clear mathematical properties that support formal analyses. Diagrams and maps help to communicate models to tool users. Plain textual representations are manly used an input for tools. An optimal model representation is probably a hybrid of these approaches.

The TRE_SPASS model is XML-based; this textual representation can be visualised by standard tools and especially the WP4 visualisations.

2.3. Infrastructure

The vast majority of the studies that represent infrastructure (16 out of 20) cover in their modelling approach the digital layer; a big portion of them models both the physical and the digital world.

Ten *et al.* (Ten et al., 2010) deal with cyber-security of critical network infrastructures. The study proposes a supervisory control and data acquisition security framework with four major components: real-time monitoring, anomaly detection, impact analysis and mitigation strategies (RAIM). Xie *et al.* (Xie et al., 2009) focus on analysing network security vulnerabilities, therefore they consider the digital layer of an infrastructure. Shahriari et al. (Shahriari et al., 2010) apply an actor-based language using reactive objects (REBECA). The study deals with network security on the Transport Protocol Layer. It models a typical network including client and server. Aiming to diagrammatically represent complex hacker attacks from multiple perspectives, the Hacker Attack Representation Method (HARM) by Karpati *et al.* (Karpati et al., 2011) uses a combination of six modeling techniques. With the help of the Misuse Case Maps (MUCM) the system architecture targeted by a specific attack is modeled. Dragovic *et al.* (Dragovic & Crowcroft, 2004, 2005; Dragovic, 2006) work in the field of information security and privacy protection in ubiquitous computing. They model the world unifying the physical and the virtual realms. Each instance in this world belongs to a container class, which can be physical, intermediate, and virtual. The notion of infrastructure in ExASyM (Probst et al., 2007; Probst & Hansen, 2008; Probst & Hansen, 2009; Sameer, 2011) is represented as set of locations and connections. The physical layer describes the architectural plan of the organisation being modelled, e.g., how rooms are connected with each other. Similar to the physical layer, ExASyM models network components and the connections between them. The MsAMS modelling framework (Franqueira et al., 2009) focuses on modelling networks and is based on ambients that represent hosts, services, vulnerabilities, networks, users, and even credentials. The social layer is also described as ambients interacting with each other. Mathew *et al.* (Mathew et al., 2005, 2008) model information about the physical location and reachability of information assets on a network. Even though the study is focused on network security, it considers both the network infrastructure and physical aspects. Samarji *et al.* (Samarji et al., 2013) focus only on modelling system networks. Sommestad *et al.* (Sommestad et al., 2010, 2012) model information systems.

Only four of the modeling approaches in this literature review reflect all three types of infrastructure in their studies. Pieters et al. (Pieters, 2011) model the physical and digital infrastructures and also reason about access in system models including human actions, and their graph-based reference model reflects all physical, digital and social infrastructures. Scott *et al.* (Scott, 2004) model the world as a nested tree of entities, similar to ambients in the Ambient Calculus. Sorts are used as contrains for how entities could be nested. The authors take into account the physical world, the digital world, though modelled as physical objects, as well as the actors, modelled as autonomous physical entities. In Portunes (Dimkov et al., 2010; Dimkov, 2012) the world is also divided in the physical, digital, and social layer. A later study by Pieters *et al.* (Pieters et al., 2013) is focused on

alignment of policies from different domains: access control, network layout, and physical infrastructure as well as the social domain. While the study is not focused explicitly on modeling these domains, they are still part of it as components of the policies being aligned.

The only socio-technical model that focuses exclusively on the physical domain is the STS model by Lenzini et al. (Lenzini, Mauw, & Ouchani, 2015). This approach models the infrastructure as a graph structure that gives rise to a labelled transition system (LTS) capturing the infrastructure state, and evaluates security properties directly on this LTS.

2.3.1. Discussion

All of the studies directly addressing modelling approaches are able to model the digital layer of the infrastructure. About half of them model the physical infrastructure as well. Only few studies reflect the social layer of the system infrastructure. In addition, most of the studies model only the attacker who traverses the system acquiring knowledge and/or access on the way.

The TRE$_S$PASS model needs to model all three layers, so it will be able to adapt to the existing approaches.

2.4. Assets and Containment

ExASyM (Probst et al., 2007; Probst & Hansen, 2008; Probst & Hansen, 2009) considers the objects that the actors work with or any data in general, be it located at actors or accessible at certain locations. Pieters *et al.* (Pieters et al., 2013) consider the assets of an organisation described by high-level policies ("sales data should not leave the organization") as well as desirable and undesirable states of those assets (being in the hands of competitors). In low level policies individual actions of actors are constrained ("this door can only be opened with a specific key"). An earlier study by Pieters *et al.* (Pieters, 2011) faces an issue with the containment approach in the case when there are different domains represented (physical, digital, and social) and the physical and digital assets being modelled are combined. Assets in Portunes (Dimkov et al., 2010; Dimkov, 2012) can belong to the physical or digital domain, e.g., a usb dongle and service data. Ten et al. (Ten et al., 2010) address cyber-security of critical infrastructures, especially electrical power infrastructure, thus they consider cyber-assets of the power infrastructure including computer and communication devices installed in power plants, substations, energy control centers, etc. Xie *et al.* (Xie et al., 2009) model network resources as assets. In the study by Dragovic et al. (Dragovic & Crowcroft, 2004, 2005; Dragovic, 2006) the modelling of assets exhausts with modelling data objects. Ambients in the MsAMS modelling framework (Franqueira et al., 2009) are the key components when modelling the world thus they are abstractions, which, among others, represent also assets. The studies by Mathew *et al.* (Mathew et al., 2005, 2008) are focused on information assets in a network. They refer mostly to critical files, which are called "jewels". When modeling coordinated

attacks, the study by Samarji *et al.* (Samarji et al., 2013) allows resource sharing between attackers, therefore different assets of a system could be threatened at the same time. Sarkar *et al.* (Sarkar et al., 2014) model assets in the form of data or artifacts, and annotations. In (Sommestad et al., 2010, 2012) assets and their relation to each other are specified and risk is estimated with regards to the assets in terms of probabilities (architectural metamodel and probabilistic dependencies).

We consider containment either as the containment of an object at a location, or an object within an other object. An example of the latter is a hard disk within a PC.

Objects and actors can be modelled to be at a location. In this case, actors also can travel within the infrastructure, gaining objects or performing actions (Pieters, 2011; Probst et al., 2007; Probst & Hansen, 2008; Probst & Hansen, 2009; Dimkov et al., 2010; Dimkov, 2012; Scott, 2004). Similar to the real world, actors can only travel within the physical infrastructure. An actor can for example go to a room and get some object (Probst & Hansen, 2009), but not the bits of a digital file. However, there can be interaction with objects in the digital infrastructure (e.g. by using a computer to start a process).

The second meaning of containment is an object within an other object, whereas the relationship between objects is more in the hierarchical sense. Such kind of relationship can be modelled by some of the approaches considered (Karpati et al., 2011; Dragovic & Crowcroft, 2004, 2005; Franqueira et al., 2009; Dimkov et al., 2010; Dimkov, 2012; Scott, 2004; Lenzini et al., 2015). Examples include a room within a building or a pc within a room, as well as the containment of a digital object in physical object (Dragovic & Crowcroft, 2004, 2005; Dimkov et al., 2010; Dimkov, 2012; Scott, 2004). Clearly this containment of digital objects in physical objects cannot be reversed, that is data objects can not contain physical objects. Another approach is to model everything as an ambient (Franqueira et al., 2009) and use nesting. A company network would be an ambient, containing other ambients, such as PCs, firewalls and network routers.

2.4.1. Discussion

All models support assets on the different levels considered. Some studies only consider information, all other include physical and to some extent digital artefacts.

Containment is only addressed by a few studies. The studies that are able to model an actor at a location, are also able to model the nesting of objects. Some studies provide a different notion of containment in the sense of annotation/quantification properties, e.g., hosts containing vulnerabilities.

The TRE$_S$PASS model can represent assets such as data and items. The model represents containment through location attributes; containment can be arbitrarily deep.

2.5. Processes, Actions, and Behaviour

Processes are generally defined as a sequence or flow of steps or actions. In the context of socio-technical modelling this is a sequence of attack steps (Karpati et al., 2011; Xie et al., 2009; Samarji et al., 2013; Pieters et al., 2013; Sommestad et al., 2010, 2012; Zhao, Huang, Jin, & Zhang, 2011; Franqueira et al., 2009) or the (data) flow through a system or application (Karpati et al., 2011; Shostack, 2008; Sarkar et al., 2014).

Processes are represented as part of a model in ExASyM (Probst et al., 2007; Probst & Hansen, 2008; Probst & Hansen, 2009; Sameer, 2011), Portunes (Dimkov et al., 2010; Dimkov, 2012) and Scott *et al.* (Scott, 2004). In the model of Scott, software checkers (e.g. Promela) ensure that processes are being free of deadlocks, race conditions and that liveness properties hold.

By using attack trees, processes are used in a different way. The path through the tree is a sequence of attack steps and therefore an attack path can be seen as a process (Ten et al., 2010). Extending the tree with Markov Processes ensure that succeeding attack steps are executed (Kriaa et al., 2012).

Actions Activities related to computing can be put in the environment of an Ambient, including hosts, services, vulnerabilities, networks, users and credentials (Franqueira et al., 2009). The activities of mobile agents that react to changes in the context is described by (Scott, 2004), actions involving mobile agents are expressed in: Out, In, Read, Eval and NewLoc (De Nicola et al., 1998).

Regarding vulnerabilities to the system, these are described in (Ten et al., 2010; Xie et al., 2009; Zhao et al., 2011). Specific intrusion sequences, including interactions and message sequences are used in (Karpati et al., 2011).

Behaviour The expression of human behaviour in general is described in terms of actions (Scott, 2004; Pieters, 2011) and how the user interacts with the system and what processes are involved (Karpati et al., 2011).

Meta-attacks are described as attacker behaviour on a system, e.g., database searches or unusual file deletion (Mathew et al., 2008) or the expected behaviour and actions an attacker must perform to achieve the goal of the attack (Viet, Panda, & Hu, 2012; Kriaa et al., 2012; Karpati et al., 2011; Vintr et al., 2012; Sommestad et al., 2010, 2012; Zhao et al., 2011). Also specific behaviour is described, e.g., actor moving between locations in a physical infrastructure (Dimkov et al., 2010; Dimkov, 2012; Probst & Hansen, 2008; Lenzini et al., 2015). The actors in the model can perform actions (e.g., change location, store data or starting a process on a computer) (Probst & Hansen, 2008), or move assets (Dimkov et al., 2010; Dimkov, 2012; Lenzini et al., 2015). Furthermore, attackers can start a processes (Shahriari et al., 2010) and it is assumed that they will pick attack steps that are related to their skills (Xie et al., 2009).

2.5.1. Discussion

All studies agree on the definition of a process as a sequence of steps. Most studies only mention the existence of processes as part of the model. However, some models provide explicit support for processes.

In most studies, actions relate to the digital domain varying from specific actions as Read and Out to “all computing” in general.

Behaviour relates to the social domain, where human actors perform actions within the model. Human behaviour involves all attacker behaviour that is needed to achieve a goal and more specific behaviour like moving between locations start or start a process on a computer. Using human behaviour as a general, non restricted concept provides flexibility. On the other hand, a restricted set of actions enables formal treatment and analysis. A trade-off should allow the freedom to model human behaviour in a proper way, and also be able to be formally checked.

Also the TRE$_S$PASS has natural support for processes through the underlying process calculus, and supports actions and behaviour through this underlying process calculus, that also encodes actor behaviour.

2.6. Actors

In ExASyM (Probst et al., 2007; Probst & Hansen, 2008; Probst & Hansen, 2009; Sameer, 2011) actors can move in the infrastructure by following the connections between the locations. In ANKH (Pieters, 2011), on the other hand, humans and non-humans are treated symmetrically. There is no need to distinguish between actors, objects, and credentials a priori. In the MsAMS framework (Franqueira et al., 2009) basically everything is represented through an ambient, including the users. Scott *et al.* (Scott, 2004) model actors as autonomous physical entities with the ability to move between rooms. Mathew *et al.* (Mathew et al., 2005, 2008) model users with different roles in order to evaluate their influence on a network and detect possible violations. Since Karpati *et al.* (Karpati et al., 2011) represent details about the actors in the system architecture in Misuse Case (MUC) diagrams, a colour notation is used to distinguish between “normal actors” (or “regular users”) and the attacker. Actors in Portunes (Dimkov et al., 2010; Dimkov, 2012) are allowed to move objects around and thus modify the graph representing the system. Actors are also able to interact with each other. Samarji *et al.* (Samarji et al., 2013) present the system in terms of predicates. The subject of an actor’s predicate is always the ID, uniquely identifying actors. Actors in DASAI (Sarkar et al., 2014) can be humans or automated agents. The agents in the system are assumed to be insiders. There is possibility to model interaction between colluding agents. In contrast, actors in (Sommestad et al., 2010, 2012) are modeled as part of the architecture, regardless of whether it is an outsider or insiders. In their work Dragovic *et al.* (Dragovic & Crowcroft, 2004, 2005; Dragovic,

2006) deal with information exposure threats where the threat does not include a malicious intruder. In the STS model actors, including the intruder, can act probabilistically and perform different actions (e.g., move or lock an object) (Lenzini et al., 2015).

2.6.1. Discussion

Most of the studies focus on insiders as they have better access and knowledge. Some of the studies do not distinguish between insiders and outsiders. Few studies do not even distinguish between humans and non-humans in the modelling phase. In real life attackers can collaborate; only few of the studies are able to represent this. One study does not even model actors or a potential attacker, but analyses the context for possible information leakage without a specific attack scenario involving an actor.

In the TRE$_S$PASS model, insiders and outsiders are the same. They differ in their knowledge about the organisation.

2.7. Policies

Low level policies manage accessibility within an infrastructure. In this sense they describe direct actions being allowed if certain conditions are satisfied.

ExASyM uses access control policies at locations. Actors need to comply with the credentials in order to be able to perform the allowed actions specified in the policy. The network attack model of Xie et al. (Xie et al., 2009) consists of attack states, attackers, and attack rules. The attack rules describe the transitions between attack states and define preconditions. For optimization purposes, Dragovic et al. (Dragovic & Crowcroft, 2004, 2005; Dragovic, 2006) assign policies to a given container class. In this way a policy applies for each instance of the class thus avoiding unnecessary repetition. Their studies deal mostly with policies concerning access control and authorization. As the world in the MsAMS framework (Franqueira et al., 2009) is based on ambients, the policies are embedded in the rules of the ambient. Samarji et al. (Samarji et al., 2013) do not define explicit policies. However, there is an implicite approach by defining predicates, modelling the assets and the knowledge of actors.

High level policies describe actions at an abstract meta-level. An example of a high level policy is "*all behaviours that have an undesirable outcome*". Pieters *et al.* (Pieters et al., 2013) focus on formally identifying misalignments between the different levels of policies, for example, access control policies and organisational ones. Scott *et al.* (Scott, 2004) consider mobility policies as well as global security policies. A potential problem of conflicting policies is encountered and a solution is proposed by describing suitable conflict resolution meta-policies. In contrast to the majority of studies in this literature review, where policies are used in order to ensure security and often attacks are derived by enforcing the policies, in this study policies are used for controlling Sentient Mobile Applications at runtime as well as making the development of such applications easier. The

Portunes modelling language (Dimkov et al., 2010; Dimkov, 2012) expresses policies from physical and digital security by low level policies, and then introduces high level policies in terms of security awareness.

2.7.1. Discussion

Most of the studies take into consideration only low level policies in terms of accessibility/reachability. While considering high level (organisational policies) is essential, it could also be problematic in case of inconsistencies between low and high level policies. One study pays close attention to this problem and derives attack scenarios from analysing the policies on different levels.

The TRE$_S$PASS model must enable the reasoning about policies and their relationship; especially contradictions between and holes in policies are of interest, since they may enable attacks.

2.8. Quantitative Measures

Quantitative measures are used to annotate model elements either during model building or as result of computations. The models can be annotated with properties related to attackers and properties related to the owners of the system. An important measure considered in studies is the probability of success of a launched attack (step) (Vintr et al., 2012; Sommestad et al., 2010, 2012). In terms of risk management, the impact of a exploited vulnerability (Zhao et al., 2011) and organisational impact (Xie et al., 2009) of attacks are of interest.

Properties considered related to attackers include annotations of monetary costs needed to perform an attack (Karpati et al., 2011; Vintr et al., 2012; Xie et al., 2009), as well as the time needed to execute an attack (Kriaa et al., 2012). Further more the needed skill for an attack (Karpati et al., 2011), vulnerability exploitability of a system (Zhao et al., 2011), and the necessity for special tools (Vintr et al., 2012). Perhaps the most valuable annotation for an attacker is the risk of detection (Karpati et al., 2011; Xie et al., 2009).

2.8.1. Discussion

A number of studies support quantitative annotations to annotate the model and attack. Those that are most frequently supported are: probability that an attack will succeed and the costs of an attack. The supported annotations are mainly properties of an attacker or actions, e.g., required skill and risk of detection, however there are also some organisational properties, e.g., impact of an attack.

The TRE$_S$PASS model supports quantitative annotations through unique identifiers for all model elements. These identifiers enable the mapping from elements to properties through the knowledge base.

2.9. Attacks, Vulnerabilities, and Countermeasures

ExASyM recognises possible attackers based on the analysis of the model and presents them as sequence of actions. Pieters *et al.* (Pieters et al., 2013) provides the basis for existing and future methods for finding security threats induced by misalignment of policies in socio-technical systems. Attacks are generated from mismatches between global policies and local ones. An attack is considered again as a sequence of actions. Ten *et al.* (Ten et al., 2010) evaluate system-, scenario-, and leaf-level vulnerabilities by identifying the system adversary objectives. In their anomaly detection they use event correlation techniques that are categorised as temporal, spatial, or hybrid. The impact analysis evaluates the consequences of cyberattacks on SCADA. Mitigation strategies introduce security improvements of the most vulnerable components of an attack scenario (presented as sequence of events). Xie *et al.* (Xie et al., 2009) present an automatic generation of attack graphs. The attack graph framework includes a host access graph and sub-attack graphs. Each individual sub-attack graph presents the attack scenarios from one specific source host to another specific target host. The host access graph presents the access relationships between each pair of hosts. Mathew *et al.* (Mathew et al., 2005, 2008) use a static analysis tool to periodically construct Capability Acquisition Graphs (CAGs) which are then analyzed to uncover any possible attacks. Information about vulnerabilities in network services is provided beforehand as an input to the tool. As the CAGs are generated periodically, there is potential for mitigation of attacks in the form of raising an alert when an unauthorized privilege accumulation becomes apparent. Shahriari *et al.* (Shahriari et al., 2010) show how an attacker can combine simple attacks into multiphase attacks. The study uses a model checker for finding counter-examples as violations. The $ST(CS)^2$ platform (Al Sabbagh & Kowalski, 2012) aims to provide its users with guided cyber security warnings based on the subscriber's socio-technical security posture. As opposed to the general cyber security warnings, which give only an overview of the current situation, the authors talk about guided security warnings where the threat level and the recommended countermeasures are customized depending on the user's socio-technical posture. In another study vulnerability is modelled in the form of possible step-wise attacks (Pieters, 2011). An attack is successful if the attacker gets access to a designated asset. Dragovic *et al.* (Dragovic & Crowcroft, 2004, 2005; Dragovic, 2006) focus on subset of information leakage threats, also called information exposure threats. In their system for autonomic context-adaptive security, they focus on reasoning about the context. The reduction of the Level of Exposure (LoE) for all data objects is achieved by two main protective actions: containment manipulation and information reduction. The vulnerabilities are provided as an input component in MsAMS modelling framework (Franqueira et al., 2009). Once the network is modelled, an attacker, who is also represented as an ambient as all other components, is simulated dynamically. In this way an attack path is found, which is allowed by the modelled ambients and their embedded rules. Different

approaches of modelling complex attacks from different perspectives are used in HARM modelling technique (Karpati et al., 2011). The study provides an integrated view of security attacks and system architecture - misuse case maps and misuse sequence diagrams. In Portunes (Dimkov et al., 2010; Dimkov, 2012) attacks are generated by finding inconsistencies between the security policies in the different domains (physical, digital and social). Respectively, an attack scenario could combine physical, digital and social means of achieving his/her goals. Samarji *et al.* (Samarji et al., 2013) derive from the model individual, coordinated (simultaneous) and concurrent attacks. There are also types of attackers' collaboration: load accumulation, load distribution, role distribution. The study formally describes attacks by presenting the system state in terms of predicates. The authors have chosen a pessimistic approach: in coordinated attacks, if a given knowledge is required, it is enough that one of the actors has this knowledge. In the study by Sommestad *et al.* (Sommestad et al., 2010, 2012) vulnerabilities are threats defined by domain experts as part of the model (both the abstract and the concrete). An abstract model is defined as a base for a concrete model. Additionally a metamodel is associated with a probabilistic model for evaluating the security risk. Countermeasures are modeled with the aim to minimize the risk. The study is focused on monetary loss from assets, but other application domains are also possible. In Sarkar *et al.* (Sarkar et al., 2014) the vulnerabilities are defined by domain experts and serve as an input to the analysis tool. An attack model is first made by a domain expert. Attack A is successful on a process P when there is a mapping relation from A to P with certain conditions being satisfied, i.e. an attack is successful if there is a "similarity match" between A and P. Countermeasures work as follow: once an attack is found, improvement points in the process are automatically identified (sorted by how heavily a certain step is attacked). P is then evaluated to check whether the improvement was successful. The STS model allows to evaluate security properties, such as the minimal cost or the maximal probability of the intruder reaching a sensitive location or object, using the probabilistic model checker PRISM (Lenzini et al., 2015).

2.9.1. Discussion

Only few of the studies provide vulnerabilities as input to the model. Usually they are described by domain experts and are based on previous attacks. The most frequently used attack representation is a sequence of actions. While it gives a high overview and is easy to understand by non-experts, it has a flat structure and thus does not provide many details. The other popular representation are attack trees, presenting threads in a hierarchical structure; however, they loose the sequential notion. Another disadvantage of attack trees (or in some studies referred to as attack patterns) is that they cannot reflect interactions between different attacks. Only two studies model attacks carried out by more than one attacker, i.e., by collaborating attackers.

The TRE$_S$PASS model represents vulnerabilities and counter measures through qualitative properties in the knowledge base.

Study	Representation	*Physical* Infrastructure	*Digital* Infrastructure	*Social* Infrastructure	Assets	Containment	Processes	Actions	Behaviour	Actors	*Low Level* Policies	*High Level* Policies	Quantitative Measures	Attacks	Vulnerabilities	Countermeasures
(Ten et al., 2010)	ADtree	-	+	-	+	-	+	+	-	-	-	-	+	+	+	+
(Xie et al., 2009)	Graph	-	+	-	+	-	+	-	-	-	+	-	+	+	+	-
(Shahriari et al., 2010)	Text	-	+	-	+	-	-	+	-	-	-	-	-	+	-	-
(Karpati et al., 2011)	Multi	-	+	-	+	-	+	+	+	+	-	-	+	+	+	-
(Dragovic & Crowcroft, 2004, 2005; Dragovic, 2006)	Tree	+	+	-	+	+	-	-	-	-	+	-	+	+	-	+
(Probst et al., 2007; Probst & Hansen, 2008)	Graphical	+	+	+	+	-	+	+	+	+	+	-	-	+	-	+
(Franqueira et al., 2009)	Multi	+	+	+	+	+	-	+	+	+	+	-	+	+	-	-
(Mathew et al., 2005, 2008)	DAG	+	+	-	+	+	-	-	-	+	+	-	-	±	-	±
(Samarji et al., 2013)	Graph	-	+	-	+	-	±	-	-	+	±	-	-	+	-	-
(Sommestad et al., 2010, 2012)	UML like	-	+	-	+	-	+	+	-	-	-	-	+	+	+	+
(Scott, 2004)	Tree	+	+	+	-	+	+	+	-	+	+	-	-	-	-	-
(Dimkov et al., 2010; Dimkov, 2012)	DAG	+	+	+	+	+	+	+	+	+	+	+	-	+	-	-
(Pieters et al., 2013)	Text + Venn	+	+	+	+	-	+	-	+	-	+	+	-	+	-	-
(Sarkar et al., 2014)	DAG	-	-	+	+	-	+	-	-	+	-	-	-	+	+	-
(Al Sabbagh & Kowalski, 2012)	UML like	-	-	-	+	-	±	+	+	+	-	-	-	+	-	+
(Pieters, 2011)	HyperGraph	+	+	+	+	+	-	+	+	+	+	-	-	+	-	-
(Zhao et al., 2011)	Graph	-	-	-	-	-	-	-	-	-	-	-	+	+	+	-
(Kriaa et al., 2012)	Graph	-	-	-	+	-	±	-	-	-	-	-	+	+	-	-
(De Nicola et al., 1998)	Text	+	+	-	+	-	+	+	-	+	-	-	-	-	-	+
(Vintr et al., 2012)	Tree	+	±	±	-	-	-	+	-	-	-	-	+	+	±	-
(Santhi, Yan, & Eidenbenz, 2010)	Graphical	-	±	-	+	+	-	+	+	-	-	-	+	+	+	-
(Maiden, Jones, Manning, Greenwood, & Renou, 2004)	UML like	-	-	+	+	-	-	+	+	+	±	±	-	-	-	-
(Franch, 2006)	Graph	-	-	+	+	-	-	+	+	-	-	-	+	-	-	-
(Mohaghegh, Kazemi, & Mosleh, 2009; Mohaghegh, 2010)	Graph	-	-	+	-	-	+	+	+	+	-	-	+	±	±	±

Table 2.1.: Models and their characteristics, one models can contain multiple references

3. Requirements for Socio-Technical Security Models

In this chapter, we summarise our findings identified for socio-technical security models through the structured literature review and our work in the TRE$_S$PASS project, as well as TRE$_S$PASS-specific requirements. In risk assessment processes, models play the role of a central data repository, that provides abstraction and storage of data about the organisation under scrutiny, provides input data for analyses, processes, and visualisations, and provides means of feeding analysis results back into the model.

The TRE$_S$PASS-specific requirements come from other work packages. In order to coordinate the requirements gathering process between the work packages, a special task force was established within the TRE$_S$PASS project. The task force collected and synchronised all the dependencies between the work packages, and a big central requirements table was created. All requirements involving WP1 are presented in Appendix B, including requirements that WP1 has to other work packages (Section B.1). The complete requirements table for the TRE$_S$PASS project can be found in (The TRE$_S$PASS Project, D6.2.2, 2015).

3.1. Representation

A central decision in developing models is the underlying representation. While in general most representations can be translated in other forms, it is essential to choose an appropriate core representation. Where necessary, this can be extended with extractors to transform it to the specific needs of certain tools and phases.

R88 The representation of socio-technical security models must provide the necessary abstraction to make the model accessible to both tools and human readers.

- **R89** The representation of the socio-technical security model must provide structured access for formal analysis methods.
- **R90** The model should support hierarchies and types of artefacts to ease modelling.
- **R91** Each element in the model should have a unique identifier.
- **R20** The model should support macros to ease modelling of, e.g., complex repeated properties and domain-specific extensions.

R92 A reader not intimately familiar with the modelling language should be able to gain a high-level understanding of the model with minimum explanations.

R93 The modelling language should support domain experts, with little or no specific training, by allowing them to quickly develop initial high-level models and analyses.

R11 There must exist a clear description of the models.

R48 It must be possible to add comments to model representations.

3.1.1. The TRE$_S$PASS Model

For the TRE$_S$PASS model these requirements implied that we choose a graph-based format that is internally represented as XML, thereby enabling wide tool support, easy transformation to different formats, and relatively painfree understanding by humans. In the XML format, type attributes are used to represent hierarchies of elements, and to associate elements with properties, and all elements are required to have a unique id that is enforced by XML parsers.

3.2. Infrastructure, Assets, and Actors

To represent socio-technical aspects of organisations, the models we consider must be able to contain physical, digital, and social aspects of organisations. These aspects include policies, processes, containment of assets, actions of processes, and behaviour of actors. The way how these elements are represented is directly influenced by the choices made for the representation described in requirement group *Representation* in Section 3.1.

While the concepts of infrastructure, assets, and actors are closely related in the studies discussed in the previous chapter, models can contain all or a subset of them. We therefor group each of them into a group of requirements, and split actors into (digital) processes and (human) actors.

It should be noted that some of the requirements are based on our current understanding of socio-technical security models. For example, the categories for assets (tangible or intangible), or the separation between processes and actors, might be too broad or too narrow for some domains.

3.2.1. Infrastructure

R19 Socio-technical security models must provide the necessary abstraction to represent infrastructure and other components.

R94 The model must be able to represent locations in the infrastructure and (directed and undirected) connections between them.

R95 To support formal methods and improve their precision, the model should support different domains for locations.

R96 The model should support creation and destruction of locations in some domains.

R21 The model needs to be able to manage a list of relevant properties of the entities (products and services) involved.

R43 The model and data extraction must be able to cope with a dynamic system, i.e., changes in the system need to be detected and represented in the model. And the model should have rules to change the model itself.

3.2.2. Assets and Containment

R97 Socio-technical security models must provide the necessary abstraction to represent assets.

R98 Assets must be able to be located at locations in the infrastructure, at actors, or at other assets.

R99 The model should support tangible and intangible assets (items and data).

R100 The model must be able to express containment between assets.

R22 The model should represent relations between different entities.

3.2.3. Processes, Actions, and Behaviour

R101 Socio-technical security models must provide the necessary abstraction to represent processes and their actions.

R102 Processes must be able to access intangible assets.

R103 Processes must be able to move in the location domain they belong to.

R104 The assets associated with a process must move with them.

R105 Processes should be able to contain a series of actions that represent the process.

R44 Actions in processes should accept wild cards instead of concrete parameters.

R45 There must be a non-destructive read action.

R24 Risks and relations should include contractional agreements and jurisdictional requirements.

3.2.4. Actors

R106 Socio-technical security models must provide the necessary abstraction to represent actors and their behaviour.

R107 Actors must be able to access tangible and intangible assets.

R108 Actors must be able to move in the location domain they belong to.

R109 The assets associated with an actor must move with them.

3.2.5. The TRE$_S$PASS Model

Since the TRE$_S$PASS model has been chosen to be graph-based, many infrastructure requirements are fulfilled automatically. Other requirements such as **R96** are partly fulfilled by the underlying semantics, which is KLAIM based. In the TRE$_S$PASS model we distinguish between physical locations, such as rooms, that can not be destroyed, and locations representing assets such as computers, that can. Destruction of these locations is modelled as input of that location.

The TRE$_S$PASS model supports items and data. The latter are represented as KLAIM-like tuples, and can therefore be located at any location. Items are represented as locations, and can themselves be located at any location. This fulfils automatically also the requirement for containment identified in the previous chapter.

Processes in the TRE$_S$PASS model are represented as locations, unlike in KLAIM, and consist of a series of KLAIM actions. Being locations, processes can store data, but access to items is prohibited by the model's semantics. Actors are also represented as processes and consequently as locations, and can store both items and data.

3.3. Policies

R110 Socio-technical security models must provide the necessary abstraction to represent policies.

R111 Policies must require a set of credentials and enable a set of actions, both possibly complete or empty.

R112 Policies should be associated with data and/or locations.

R46 Policies should contain variables to bind parameters in the enabled actions to concrete values in credentials, or to enforce certain values.

R47 Global Keywords to provide metainfo in policies or processes.

3.3.1. The TRESPASS Model

Policies are a crucial part of TRESPASS models, since they guide the attack detection, and since they describe the necessary credentials for access to certain locations or data. TRESPASS policies do support variables, which are treated with unification of an actor's assets.

3.4. Quantitative Measures

The data about socio-technical security models, especially the social layer of it, needs to be made available to analyses, processes, and visualisations through the model. Some of this data will be specific to a certain element in the model, other will depende on the type, or some contextual information.

R113 Socio-technical security models must support the storage of data for elements in the model.

R114 The data must be retrievable and settable, based on the element, its type, or a certain action to be performed on it.

3.4.1. The TRESPASS Model

All elements in the TRESPASS model have a type and a unique identifier (c.f., **R90** and **R91**). This information makes it possible to retrieve specific data for this element. The same holds if a process or actor wants to perform an action on some artefact, e.g., wants to social engineer another actor. The model does not specify how to store this data, but interfaces, e.g., to a database that provides the data collected through the tools and processes in WP2.

3.5. Attacks, Vulnerabilities, and Countermeasures

The model must be able to represent the result of analyses performed on the model. These analyses can result in identification of attacks or vulnerabilities, or can lead to introduction of countermeasures.

R115 Socio-technical security models must support the storage of analysis results in the model.

R116 The model must support to associate model elements with attacks and vulnerabilities.

R117 The model must support locations that represent countermeasures.

3.5.1. The TRESPASS Model

Both requirements **R116** and **R117** can be realised similar to quantitative measures, where the unique identifiers of model elements are mapped to the attacks that they contribute to, or where the relevant quantitative measure is adapted based on the vulnerability identified. Physical countermeasures, which result in a change of measure, e.g., a longer time to success if a stronger door is used, or that increase access control, can be represented by a location being added to the model.

3.6. Programmatic Interface

A consequence of the requirements on representation above is that we need an API to establish the interface between a GUI and the model, and the model and analyses, processes, and visualisations.

R118 There should exist an API to the model that supports access to model elements and data associated with these elements.

- **R85** There must be a method to create an empty model.
- **R66** The system can retrieve standard model components by name.
- **R51** Navigator maps can be automatically generated from an ArchiMate model.
- **R119** Model elements must be accessible by their identifier.

3.6.1. The TRESPASS Model

The TRESPASS model API is used by the tools and analyses, and provides the necessary methods to obtain elements by type, identifier, or connections.

4. Conclusions

This deliverable presents the final requirements identified in the TRE$_S$PASS project for socio-technical security models. These requirements have guided the development of these models in the project, and we believe they provide valuable guidance to developers of similar models.

References

Al Sabbagh, B., & Kowalski, S. (2012, June). ST(CS)2 - Featuring socio-technical cyber security warning systems. In *Proceedings title: 2012 international conference on cyber security, cyber warfare and digital forensic (cybersec)* (pp. 312–316). IEEE. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6246110 doi: 10.1109/CyberSec.2012.6246110

Bettini, L., Loreti, M., & Pugliese, R. (2002). An infrastructure language for open nets. In *Proceedings of the 2002 acm symposium on applied computing* (pp. 373–377). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/508791.508862 doi: 10.1145/508791.508862

De Nicola, R., Ferrari, G., & Pugliese, R. (1998). Klaim: a kernel language for agents interaction and mobility. *Software Engineering, IEEE Transactions on*, *24*(5), 315–330. doi: 10.1109/32.685256

Dimkov, T. (2012). *Alignment of organizational security policies – theory and practice* (Unpublished doctoral dissertation). University of Twente.

Dimkov, T., Pieters, W., & Hartel, P. H. (2010, March). Portunes: representing attack scenarios spanning through the physical, digital and social domain. In *Proceedings of the joint workshop on automated reasoning for security protocol analysis and issues in the theory of security (arspa-wits'10). revised selected papers, paphos, cyprus* (Vol. 6186, pp. 112–129). Berlin: Springer Verlag. http://eprints.eemcs.utwente.nl/17295/.

Dragovic, B. (2006). Casper: Containment-aware security for pervasive computing environments. *Doctor of philosophy, St John's College, University of Cambridge (March 2006)*.

Dragovic, B., & Crowcroft, J. (2004). Information exposure control through data manipulation for ubiquitous computing. In *Nspw* (pp. 57–64).

Dragovic, B., & Crowcroft, J. (2005). Containment: from context awareness to contextual effects awareness. In *Proceedings of 2nd inernational workshop on software aspects of context. ceur workshop proceedings.*

Franch, X. (2006). On the quantitative analysis of agent-oriented models. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4001 LNCS*, 495-509. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-33746064822&partnerID=40&md5=450a61eb334820b1ee26768fd476dc1c (cited By 14) doi: 10.1007/11767138_33

Franqueira, V. N. L., Lopes, R. H. C., & van Eck, P. (2009). Multi-step attack modelling and simulation (msams) framework based on mobile ambients. In *Proceedings of the 2009 acm symposium on applied computing* (pp. 66–73). New York, NY, USA:

ACM. Retrieved from http://doi.acm.org/10.1145/1529282.1529294 doi: 10.1145/1529282.1529294

Gorla, D., & Pugliese, R. (2003). Resource access and mobility control with dynamic privileges acquisition. In *In proc. of icalp'03, volume 2719 of lncs* (pp. 119–132). Springer-Verlag.

Hunker, J., & Probst, C. W. (2011). Insiders and insider threats an overview of definitions and mitigation techniques. *Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications*.

Karpati, P., Opdahl, A. L., & Sindre, G. (2011). *HARM: Hacker Attack Representation Method.* doi: 10.1007/978-3-642-29578-2_10

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*, 2004.

Kriaa, S., Bouissou, M., & Pietre-Cambacedes, L. (2012, October). Modeling the Stuxnet attack with BDMP: Towards more formal risk assessments. In *2012 7th international conference on risks and security of internet and systems (crisis)* (pp. 1–8). IEEE. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6378942 doi: 10.1109/CRISIS.2012.6378942

Lenzini, G., Mauw, S., & Ouchani, S. (2015). Security analysis of socio-technical physical systems. *Computers and Electrical Engineering*. Retrieved from http://www.sciencedirect.com/science/article/pii/S0045790615000671 (Available online 6 April 2015)

Maiden, N., Jones, S., Manning, S., Greenwood, J., & Renou, L. (2004). Model-driven requirements engineering: synchronising models in an air traffic management case study. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3084*, 368-383. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-35048852501&partnerID=40&md5=4f6ff8d8561fe2a935d9358fd7329641 (cited By 21)

Mathew, S., Britt, D., Giomundo, R., Upadhyaya, S., Sudit, M., & Stotz, A. (2005). Real-time multistage attack awareness through enhanced intrusion alert clustering. In *Military communications conference, 2005. milcom 2005. ieee* (pp. 1801–1806 Vol. 3). doi: 10.1109/MILCOM.2005.1605934

Mathew, S., Upadhyaya, S., Ha, D., & Ngo, H. (2008). Insider abuse comprehension through capability acquisition graphs. In *Information fusion, 2008 11th international conference on* (pp. 1–8).

Mohaghegh, Z. (2010). Combining system dynamics and bayesian belief networks for socio-technical risk analysis. *ISI 2010 - 2010 IEEE International Conference on Intelligence and Security Informatics: Public Safety and Security*, 196-201. doi: 10.1109/ISI.2010.5484736

Mohaghegh, Z., Kazemi, R., & Mosleh, A. (2009). Incorporating organizational factors into probabilistic risk assessment (pra) of complex socio-technical systems: A hybrid technique formalization. *Reliability Engineering and System Safety*, *94*(5), 1000-1018. (cited By 95) doi: 10.1016/j.ress.2008.11.006

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

http://eprints.eemcs.utwente.nl/19934/.

Pieters, W. (2011). Representing humans in system security models: An actor-network approach [Technical Report]. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *2*(1), 75–92. http://eprints.eemcs.utwente.nl/19934/.

Pieters, W., Dimkov, T., & Pavlovic, D. (2013, June). Security Policy Alignment: A Formal Approach. *IEEE Systems Journal*, *7*(2), 275–287. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6363516 doi: 10.1109/JSYST.2012.2221933

Probst, C., & Hansen, R. (2009, may). Analysing access control specifications. In *Systematic approaches to digital forensic engineering, 2009. sadfe '09. fourth international ieee workshop on* (pp. 22–33). doi: 10.1109/SADFE.2009.13

Probst, C. W., & Hansen, R. R. (2008, nov). An extensible analysable system model. *Inf. Secur. Tech. Rep.*, *13*(4), 235–246. Retrieved from http://dx.doi.org/10.1016/j.istr.2008.10.012 doi: 10.1016/j.istr.2008.10.012

Probst, C. W., Hansen, R. R., & Nielson, F. (2007). Where can an insider attack? In T. Dimitrakos, F. Martinelli, P. Y. Ryan, & S. Schneider (Eds.), *Formal aspects in security and trust* (Vol. 4691, pp. 127–142). Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-540-75227-1_9 doi: 10.1007/978-3-540-75227-1_9

Samarji, L., Cuppens, F., Cuppens-Boulahia, N., Kanoun, W., & Dubus, S. (2013). Situation calculus and graph based defensive modeling of simultaneous attacks. In *Cyberspace safety and security* (pp. 132–150). Springer.

Sameer, K. (2011). *Attack gerenation form system models* (Unpublished master's thesis). Aalto University, Finland.

Sanders, W. H., & Freire, R. S. (1993). Efficient simulation of hierarchical stochastic activity network models. *Discrete Event Dynamic Systems*, *3*(2-3), 271–299.

Santhi, N., Yan, G., & Eidenbenz, S. (2010). Cybersim: Geographic, temporal, and organizational dynamics of malware propagation. *Proceedings - Winter Simulation Conference*, 2876-2887. doi: 10.1109/WSC.2010.5678982

Sarkar, A., Kohler, S., Riddle, S., Ludäscher, B., & Bishop, M. (2014). Insider attack identification and prevention using a declarative approach. In *Security and privacy workshops (spw), 2014 ieee* (pp. 265–276).

Scott, D. (2004). *Abstracting Application-Level Security Policy for Ubiquitous Computing* (Doctoral dissertation, University of Cambridge). Retrieved from http://www.cl.cam.ac.uk/research/dtg/www/files/publications/public/djs55/approved.pdf (PhD Thesis)

Scott, D., Beresford, A., & Mycroft, A. (2003, jun). Spatial Policies for Sentient Mobile Applications. In *IEEE 4th International Workshop on Policies for Distributed Systems and Networks* (pp. 147–157). Retrieved from http://www.cl.cam.ac.uk/research/dtg/www/files/publications/public/arb33/ScottBeresfordMycroft-SpatialPolicies4MobileApps-IEEEPolicy2003.pdf (Conference Paper)

Shahriari, H. R., Makarem, M. S., Sirjani, M., Jalili, R., & Movaghar, A. (2010, September). Vulnerability analysis of networks to detect multiphase attacks using the actor-based language Rebeca. *Computers & Electrical Engineering*, *36*(5), 874–885. Retrieved

from http://linkinghub.elsevier.com/retrieve/pii/S0045790608000451 doi: 10.1016/j.compeleceng.2008.04.009

Shostack, A. (2008). Experiences threat modeling at microsoft. In *Modeling security workshop. dept. of computing, lancaster university, uk.*

Sommestad, T., Ekstedt, M., & Holm, H. (2012). The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures. *Systems Journal, IEEE*, *PP*(99), 1–1. doi: 10.1109/JSYST.2012.2221853

Sommestad, T., Ekstedt, M., & Johnson, P. (2010). A probabilistic relational model for security risk analysis. *Computers & Security*, *29*(6), 659 - 679. doi: http://dx.doi.org/10.1016/j.cose.2010.02.002

Ten, C.-W., Manimaran, G., & Liu, C.-C. (2010, July). Cybersecurity for Critical Infrastructures: Attack and Defense Modeling. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *40*(4), 853–865. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5477189 doi: 10.1109/TSMCA.2010.2048028

The TRESPASS Project, D1.1.1. (2013). *Initial specifications and requirements for socio-technical security models.* (Deliverable D1.1.1)

The TRESPASS Project, D1.3.2. (2015). *Extensibility of socio-technical security models.* (Deliverable D1.3.2)

The TRESPASS Project, D1.3.3. (2015). *Dynamic features of socio-technical security models.* (Deliverable D1.3.3)

The TRESPASS Project, D5.1.2. (2015). *Final requirements for process integration.* (Deliverable D5.1.2)

The TRESPASS Project, D6.2.2. (2015). *Final refinement of functional requirements.* (Deliverable D6.2.2)

Viet, K., Panda, B., & Hu, Y. (2012, October). Detecting collaborative insider attacks in information systems. In *2012 ieee international conference on systems, man, and cybernetics (smc)* (pp. 502–507). IEEE. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6377774 doi: 10.1109/ICSMC.2012.6377774

Vintr, Z., Valis, D., & Malach, J. (2012, October). Attack tree-based evaluation of physical protection systems vulnerability. In *2012 ieee international carnahan conference on security technology (iccst)* (pp. 59–65). IEEE. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6393538 doi: 10.1109/CCST.2012.6393538

Xie, A., Chen, G., Wang, Y., Chen, Z., & Hu, J. (2009, July). A New Method to Generate Attack Graphs. In *2009 third ieee international conference on secure software integration and reliability improvement* (pp. 401–406). IEEE. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5325344 doi: 10.1109/SSIRI.2009.32

Zhao, F., Huang, H., Jin, H., & Zhang, Q. (2011). A hybrid ranking approach to estimate vulnerability for dynamic attacks. *Computers and Mathematics with Applications*, *62*(12), 4308–4321. doi: j.camwa.2011.09.031

A. Project Summary

This chapter gives an overview of the TRE$_S$PASS project and its use cases. The section is shared by the public deliverables to provide the necessary background and to put the current deliverable in context.

Information security threats to organisations have changed completely over the last decade, due to the complexity and dynamic nature of infrastructures and attacks. Attacks like StuxNet involve technical and human factors, and they damage physical infrastructure. The recent attack on a German steel mill [1] was a combination of both targeted phishing emails and social engineering attacks. The phishing helped the hackers extract information they used to gain access to the plant's office network and then its production systems. As a result, the technical infrastructure of the mill suffered severe damage.

The attack on the German steel mill illustrates that we need to integrate the social and technical aspects of systems in assessing their security - and we need to do so today. Socio-technical systems pose new challenges by combining parts for which we often understand the security issues; the combined system is however much more complex due to interactions between these parts.

The main innovation of the TRE$_S$PASS project is the attack navigator, a tool and metaphor that enables defenders to predict and preventing attacks on socio-technical systems. The attack navigator supports current risk-assessment techniques with the TRE$_S$PASS process (developed in Work Package WP5), an analytical approach to identifying attacks and evaluating their impact.

The four main stages in the TRE$_S$PASS process are *data collection*, *modelling*, *analysis*, and *visualisation*. Data collection (WP2) is vital to understanding the nature of a scenario and providing input to subsequent tasks of modelling, analysis and visualisation. Within the project, the focus has been on collection and analysis of social, technical and physical data and the ways in which these relate to one another. Within each of these domains, different approaches have been taken to provide different viewpoints on the nature of the organisation being investigated.

The models (WP1) developed in TRE$_S$PASS can be adapted to the application scenario. We have developed physical modelling techniques in order to understand where further investigation may usefully be targeted. The TRE$_S$PASS model describes relevant aspects of the organisation and their connections. To explore contractual and commercial relationships, the e3value method has been adopted.

[1] BBC News, *Hack attack causes 'massive damage" at steel works*, http://www.bbc.com/news/technology-30575104, last visited October 31, 2015.

The analysis methods (WP3) developed in TRE$_S$PASS identify attacks in models and identify the most effective controls to prohibit these attacks. The analyses are supported by tools and together they provide the defender with a comprehensive understanding of properties attacks, *e.g.*, cost for the attacker, required skills, or required time.

The innovative visualisations (WP4) developed in TRE$_S$PASS focus focus particularly on visualising elements of the analysis, as this is key to the overall project goal of providing "decision support" to practitioners. However, visualisations contribute also to model development and data gathering.

Practitioners can access the TREsPASS toolkit via the attack navigator map interface, which provides an intuitive means of selecting appropriate tools (WP6) for data gathering, modelling, analysis and visualisation. These can be used, individually or in combination, to strengthen operational and strategic decision-making.

A.1. Case Studies

The TREsPASS process and tools are validated by means of case studies (WP7) in the area of cloud infrastructure, telecommunications infrastructure, ATM infrastructure, and an organisation processing privacy sensitive data.

A cloud infrastructure shares infrastructure within or across organisations, giving the cloud services provider and its employees full physical and logical access to all resources across the different consumers. In TRE$_S$PASS we formalise typical components in cloud infrastructures as well as human actors and their interrelationships, to identify their contribution to attacks on the organisation.

In telco infrastructure new products need to be launched under significant time pressure, often opening up loopholes for so-called knowledge insiders who know the market very well, trying to make as much monetary gain from the new products as possible. In TRE$_S$PASS we model both the infrastructure and contractual relationships to identify physical and monetary attacks.

The ATM infrastructure connects machines that are composed of a money safe and a computer that controls the ATM's devices. There are well protected ATMs installed inside bank branches, while others are deployed in the street and some are not even embedded in a wall. ATM attacks are common and include classic physical attacks and emerging digital attacks. In TRE$_S$PASS we model ATM installations, and identify attack likelihoods using geospatial data.

The organisation processing privacy sensitive data develops a system supporting primarily elderly and disabled people in performing online payments and managing their own money from their home. This case study involves from strictly technical security aspects, such as how information is protected while stored or transmitted, to socio-technical security aspects covering security issues arising from the use of and interaction with the technology. In TRE$_S$PASS we identify social-engineering and trust-based attacks on such systems.

A.2. Overview of TRESPASS Integration

The TRESPASS workflow involves several stages with various activities, some of which are optional. Figure A.2 shows the architecture diagram and Figure A.1 shows a visual description of the notation used. In practice, stages may not follow a linear order. For example, depending on the goal of the risk assessment, new data requests may be issued later in the process, or automatic updates of data may be supported.

The **Data collection stage** prepares for analysis and modelling steps, and may require the gathering of one or more of the following kinds of data.

Physical data collection provides knowledge about the physical layout of the organization including locations, buildings, rooms, doors, windows, etc.

Digital data collection gathers information about the organization's IT infrastructure.

Social data collection focuses on organisational and individual data, and results in actor profiles containing, *e.g.*, attributes of employees, stakeholders, or potential attackers.

Commercial data collection gathers information required for *e3fraud* analyses, which focus on potential fraud.

Stakeholder goal collection identifies assets and policies the protection of which is critical to one or more stakeholders.

The **model creation stage** handles the creation of the TRESPASS model and associated actor profiles. The e3value model creation process is complementary to the main TRESPASS model, for cases requiring a more specific financial focus:

TRESPASS model creation is a key activity result in a system model that can be further extended and analysed.

- **Components customization (optional)** takes place before or during the TRESPASS model creation to create specialized custom model components.
- **Attacker profile creation** creates the attacker profile that the TRESPASS model analysis should consider, based on ready-made attacker profiles.
- **Defender/target profile creation** creates similar profiles for the other actors in the model based on the social data gathered in the social data collection activity.

e3value model creation This interactive activity involves using the *e3value toolkit* [2] to create business value models. These models structure the commercial information gathered in the data collection stage in a formal way.

In the **analysis stage** different analyses are possible depending on the model chosen. The analysis of the TRESPASS model involves these steps:

1. In the **attacker profile selection**, the user selects the attacker profile to use in the analysis.

[2] http://e3value.few.vu.nl/tools/

2. The **attacker goals creation** provides the attack generation with the attacker goals. These can be derived by hand from the stakeholder goals or deduced automatically from the selected attacker profiles.
3. The **scenario selection** selects a scenario, consisting of a single pair of attacker and attacker goal, to run the TRESPASS analysis on.
4. To extend attack trees, **attack pattern creation and sharing** provides libraries with known attack steps. The attack tree generation can only reach a certain level of abstraction, which may not be sufficient for quantitative analyses.
5. **Attack generation** transforms the TRESPASS model to an attack tree.
6. **Attack tree annotation & augmentation** then extends the attack tree with attack patterns and decorates leaf nodes with parameter values from the data collection stage for quantitative analysis.
7. The **attack tree analyses** compute quantitative properties of attacks, *e.g.*, utility for the attacker or success probability of the attack.

The analysis of the **e3value model** is complementary to the core TRESPASS analysis and has only one step:

1. For the **fraud model generation**, the user needs select an attacker and an interval of expected occurrence rates of the commercial transactions specified by the e3value model. The e3fraud tool then identifies all possible violations of contracts, the loss for actors, and the delta in profit for the other actors.

The **visualisation stage** can be used continuously to provide practitioners with feedback regarding the results of their activities:

1. **Fraud model visualisation** shows the generated attacks as a ranked list of textual descriptions of the attack steps and displays charts showing the profitability for each actor.
2. **Attack tree visualisation** shows the intermediary attack trees.
3. **Attack tree analysis visualisation** visualises analysis results.

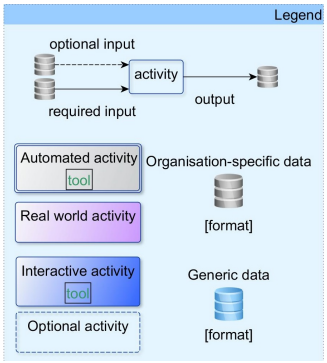

Figure A.1.: Legend for the Integration diagram in Figure A.2.

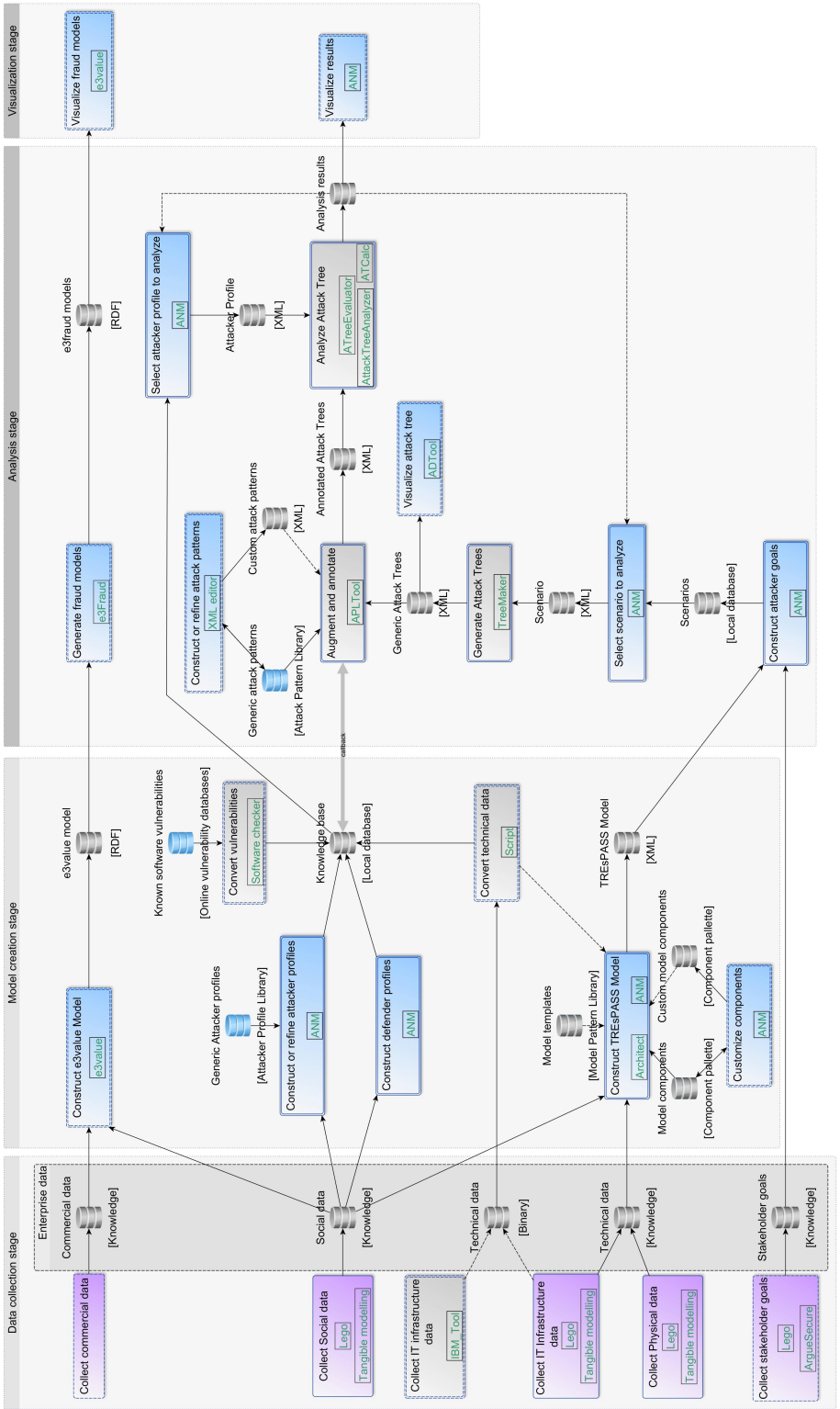

Figure A.2.: Integration diagram for the TRESPASS project.

B. Central Requirements Table for WP1

B.1. Requirements from WP1

Requirement R16

Requirement : Provide model content such as infrastructure, policies, etc
Source WP: WP1
Target WP: WP2
Goals: Needed for model
Acceptance criteria: For the case study scenarios, the relevant elements and their properties are provided as text documents, and enable a modeller to create a matching WP1 model.
Status: Agreed
Dependencies: None

Requirement R17

Requirement : Map analysis results back to model
Source WP: WP1
Target WP: WP3
Goals: Needed to communicate analysis result back to TREsPASS tools
Acceptance criteria: For a non-trivial attack, the model elements that are involved in the attack can be identified through API calls.
Status: Agreed
Dependencies: None

Requirement R18

Requirement : Programmatic interface in other components in the TREsPASS tool
Source WP: WP1
Target WP: WP6
Goals: Needed to provide access to model
Acceptance criteria: The model is successfully integrated in a workflow ANM-model-treemaker-analyses-visualisations.
Status: Agreed
Dependencies: None

B.2. Requirements to WP1

Requirement R11

Requirement : There must exist a clear description of the models.
Source WP: WP3
Target WP: WP1
Goals: Needed to produce visualisation toolkit
Acceptance criteria: A well-defined language for the socio-technical model.
Status: Agreed
Dependencies: None

Requirement R19

Requirement : Socio-technical security models must provide the necessary abstraction to represent infrastructure and other components.
Source WP: WP2,6
Target WP: WP1
Goals: Needed to use the model
Acceptance criteria: XML Format defined
Status: Agreed
Dependencies: None

Requirement R20

Requirement : The model should support macros to ease modelling of, e.g., complex repeated properties and domain-specific extensions.
Source WP: WP5-7
Target WP: WP1
Goals: Needed for case studies
Acceptance criteria: All the reasonable system components that I can think of can be modeled using the model extension framework.
Status: Agreed
Dependencies: None

Requirement R21

Requirement : The model needs to be able to manage a list of relevant properties of the entities (products and services) involved.
Source WP: WP7
Target WP: WP1
Goals: Needed in order to incorporate telco scenarios (The model should be able to represent actors, assets, policies, etc. from the real world involved in the provisioning of telecommunications products and services.)
Acceptance criteria: The model should explicitly show entities and their properties in a telco scenario.
Status: Agreed
Dependencies: None

Requirement R22

Requirement : The model should represent relations between different entities
Source WP: WP7
Target WP: WP1
Goals: Needed in order to incorporate telco scenarios
Acceptance criteria: The model should show the relations between entities in a telco scenario.
Status: Shelved
Dependencies: None

Requirement R24

Requirement : Risks and relations should include contractional agreements and jurisdictional requirements.
Source WP: WP7
Target WP: WP1,3
Goals: Needed in order to incorporate telco scenarios
Acceptance criteria: For WP1: Should model the contractual agreements and jurisdictional requirements between entities. For WP3: generation of attacks (risks) considering the contractual agreements and jurisdictional requirements between entities.
Status: Completed
Dependencies: None

Requirement R43

Requirement : The model and data extraction must be able to cope with a dynamic system, i.e., changes in the system need to be detected and represented in the model. And the model should have rules to change the model itself.
Source WP: WP7
Target WP: WP1,2
Goals: Needed for cloud case study, since the systems are highly dynamic.
Acceptance criteria: The data extraction tools need to detect and obtain change events in the cloud infrastructure. Those change events need to be translated into a change for the model and applied to the model.
Status: Agreed
Dependencies: None

Requirement R44

Requirement : Actions in processes should accept wild cards instead of concrete parameters.
Source WP: WP7
Target WP: WP1
Goals: Allowing masked IP addresses in processes was necessary to simulate packet forwarding. This operation is indicated by the operator tilde.
Acceptance criteria: Model needs to be able to express wildcards such as in IP addresses. Tools need to able to parse it.
Status: Agreed
Dependencies: None

Requirement R45

Requirement : There must be a non-destructive read action.
Source WP: WP7
Target WP: WP1
Goals: In order to implement knowledge assets on systems, a non-destructive read operation is very useful. For example, FTP or HTTP operations can be encoded as simple read calls to the tuplespace, rather than including separate processes for every available asset. In addition to brevity, this representation also allows for assets to be moved or deleted.
Acceptance criteria: The model needs to provide syntactic sugar to express non-destructive read operations in a consice way.
Status: Agreed
Dependencies: None

Requirement R46

Requirement : Policies should contain variables to bind parameters in the enabled actions to concrete values in credentials, or to enforce certain values.
Source WP: WP7
Target WP: WP1
Goals: Policies can use free variables, as in the following example: policies = { [X, contains(friends,X)] : {out(""request"", fileX, X)}} processes = { in(""request"", !F, !src).out(F)@src } The left hand side of this policy is consistent with previous descriptions of this format, allowing that if a request originates from actor X, and X is in the set friends, then the operation is permitted. Restricting the contents of the out operation to start with ""request"", fileX was also supported. The extension is to allow the free variable X also to appear on the right hand side.
Acceptance criteria: The model and tools needs to be able to handle free variables, for instance, as in the given example.
Status: Agreed
Dependencies: None

Requirement R47

Requirement : Global Keywords to provide metainfo in policies or processes.
Source WP: WP7
Target WP: WP1
Goals: A number of policy left hand side elements are implicit, such as a plain reference to an actor or location requiring that the request be originated by that actor or in that location. Rather than extend this to functions, such as using knows(info) to indicate that the actor that originates the operation must know asset info, the global keyword ACTOR has been Introduced to refer to the actor that originated an operation. Using this keyword, the above example becomes knows(ACTOR, info), which is consistent with other uses of the knows function.
Acceptance criteria: The model and tools needs to be able to handle global keywords, for instance, as in the given example.
Status: Agreed
Dependencies: None

Requirement R48

Requirement : It must be possible to add comments to model representations.
Source WP: WP7
Target WP: WP1
Goals: Comments can be used to quickly disable parts of the model but also to add explanations. Especially with automatically-generated files, explanations are often helpful.
Acceptance criteria: The model and tools need to be able to handle comments, at least line comments.
Status: Completed
Dependencies: None

Requirement R51

Requirement : Navigator maps can be automatically generated from an ArchiMate model.
Source WP: MT
Target WP: WP1
Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 1 (Security Investment) and Use-case 2 (Audit), specifically U1.4 , U4.1 and U5.8.
Acceptance criteria: 80perc. of these users accept the risk results derived from ArchiMate based TRESPASS models.
Status: Agreed
Dependencies: None

Requirement R66

Requirement : The system can retrieve standard model components by name.
Source WP: MT
Target WP: WP1
Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 2 (Audit) and Use-case 5 (Quick scan), specifically U2.5, U5.2 and U5.7
Acceptance criteria: In 80perc. of the retrievals, the user is satisfied with the contents of the selected component
Status: Agreed
Dependencies: None

Requirement R84

Requirement : The TRESPASS analysis should suggest a ranked list of countermeasures, with associated map components which can be dragged-and-dropped into the model and affect the analysis accordingly. The countermeasures should be available in a library.
Source WP: MT
Target WP: WP1,WP2,WP3,WP4
Goals: Supports Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U2.10, U2.12, U.4.2, U5.3, U5.13, U5.14.
Acceptance criteria: 80perc. of the users specified in the Use Cases are satisfied with the suggested countermeasures. 80perc. of users are able to successfully include counter- measures them into the model.
Status: Shelved
Dependencies: None

Requirement R85

Requirement : There must be a method to create an empty model.
Source WP: MT
Target WP: WP1
Goals: Supports Use-case 1 (Security Investment) and Use-case 3 (Innovation), specifically U1.3 and U3.1
Acceptance criteria: There must be a method to create an empty model in the API.
Status: Completed
Dependencies: None

B.3. Requirements within WP1

Requirement R88

Requirement : The representation of socio-technical security models must provide the necessary abstraction to make the model accessible to both tools and human readers.
Source WP: WP1
Target WP: WP1
Goals: Needed to support both tools and enable simple manual checks.
Acceptance criteria: The TREsPASS tools are able to work with the model, and the users in the case studies can summarize the structure defined by a given text model.
Status: Completed
Dependencies: None

Requirement R89

Requirement : The representation of the socio-technical security model must provide structured access for formal analysis methods.
Source WP: WP1
Target WP: WP1
Goals: Needed to support the interaction between analysis and model.
Acceptance criteria: The analyses developed in WP3 are able to obtain the necessary input.
Status: Agreed
Dependencies: None

Requirement R90

Requirement : The model should support hierarchies and types of artefacts to ease modelling.
Source WP: WP1
Target WP: WP1
Goals: Needed to model some aspects of containment.
Acceptance criteria: 80perc of the users specified in the case studies use hierarchies when modelling their case study where applicable.
Status: Agreed
Dependencies: None

Requirement R91

Requirement : Each element in the model should have a unique identifier.
Source WP: WP1
Target WP: WP1
Goals: Needed to retrieve elements and to ease refering to unique model elements.
Acceptance criteria: Model elements can be accessed by a unique identifier
Status: Completed
Dependencies: None

Requirement R92

Requirement : A reader not intimately familiar with the modelling language should be able to gain a high-level understanding of the model with minimum explanations.
Source WP: WP1
Target WP: WP1
Goals: The modelling language should be selfexplaining to enable quick sanity checks of model files.
Acceptance criteria: 80perc of the users specified in the case studies can summarize the structure defined by a given text model.
Status: Completed
Dependencies: None

Requirement R93

Requirement : The modelling language should support domain experts, with little or no specific training, by allowing them to quickly develop initial high-level models.
Source WP: WP1
Target WP: WP1
Goals: The lower the initial learning curve, the faster the uptake of TREsPASS will be.
Acceptance criteria: 80perc of the users specified in the case studies can after some training use the TREsPASS tools to define a small, high-level model.
Status: Agreed
Dependencies: None

Requirement R94

Requirement : The model must be able to represent locations in the infrastructure and (directed and undirected) connections between them.
Source WP: WP1
Target WP: WP1
Goals: Infrastructure is an essential component of socio-technical systems.
Acceptance criteria: Locations that are present in the case studies can be represented in the according model.
Status: Completed
Dependencies: None

Requirement R95

Requirement : To support formal methods and improve their precision, the model should support different domains for locations.
Source WP: WP1
Target WP: WP1
Goals: Domains allow to avoid impossible analysis artefacts, for example, a human actor entering a computer.
Acceptance criteria: Locations that are in different domains can be assigned to such when modelling the scenario.
Status: Completed
Dependencies: None

Requirement R96

Requirement : The model should support creation and destruction of locations in some domains.
Source WP: WP1
Target WP: WP1
Goals: For certain domains such as cloud infrastrcutures, model elements must be created from insider the model, eg, for virtual machine creation.
Acceptance criteria: For relevant case studies, the creation and desctruction of locations can be modelled, for example, for the cloud case study.
Status: Completed
Dependencies: None

Requirement R97

Requirement : Socio-technical security models must provide the necessary abstraction to represent assets.
Source WP: WP1
Target WP: WP1
Goals: Assets are an essential component of socio-technical systems.
Acceptance criteria: Relevant assets that occur in the case studies can be represented in the model.
Status: Completed
Dependencies: None

Requirement R98

Requirement : Assets must be able to be located at locations in the infrastructure, at actors, or at other assets.
Source WP: WP1
Target WP: WP1
Goals: The model must be able to represent where assets are located to make them accessible to analyses.
Acceptance criteria: The location of relevant assets can be represented in the model.
Status: Completed
Dependencies: None

Requirement R99

Requirement : The model should support tangible and intangible assets (items and data).
Source WP: WP1
Target WP: WP1
Goals: Not all assets are tangible but available as knowledge or data. This is so large a group, that it deserves a categorie of its own.
Acceptance criteria: Relevant data and items that occur in the case studies can be represented in the model.
Status: Completed
Dependencies: None

Requirement R100

Requirement : The model must be able to express containment between assets.
Source WP: WP1
Target WP: WP1
Goals: Containment enables us to model data in a hard disk in a compute.
Acceptance criteria: Assets that are located at assets can be represented in the model.
Status: Agreed
Dependencies: None

Requirement R101

Requirement : Socio-technical security models must provide the necessary abstraction to represent processes and their actions.
Source WP: WP1
Target WP: WP1
Goals: Processes and actions are an essential component of socio-technical systems.
Acceptance criteria: Relevant processes that occur in the case studies can be represented in the model.
Status: Agreed
Dependencies: None

Requirement R102

Requirement : Processes must be able to access intangible assets.
Source WP: WP1
Target WP: WP1
Goals: The model must be able to represent processes handling data.
Acceptance criteria: Relevant processes that work with data can be represented in the model.
Status: Agreed
Dependencies: None

Requirement R103

Requirement : Processes must be able to move in the location domain they belong to.
Source WP: WP1
Target WP: WP1
Goals: Domains reduce analysis artefacts by avoiding processes that enter building locations, for example.
Acceptance criteria: Moving processes are restricted to their domain.
Status: Agreed
Dependencies: None

Requirement R104

Requirement : The assets associated with a process must move with them.
Source WP: WP1
Target WP: WP1
Goals: Processes do not forget data when moving.
Acceptance criteria: For relevant processes the data they "know" is still available once they moved.
Status: Agreed
Dependencies: None

Requirement R105

Requirement : Processes should be able to contain a series of actions that represent the process.
Source WP: WP1
Target WP: WP1
Goals: Actions define processes, so these are essential.
Acceptance criteria: For processes that are put in the model, their functionality based on some ground actions, can be represented in the model.
Status: Agreed
Dependencies: None

Requirement R106

Requirement : Socio-technical security models must provide the necessary abstraction to represent actors and their behaviour.
Source WP: WP1
Target WP: WP1
Goals: Actors are an essential component of socio-technical systems.
Acceptance criteria: Relevant actors and their behaviour can be represented in the model
Status: Agreed
Dependencies: None

Requirement R107

Requirement : Actors must be able to access tangible and intangible assets.
Source WP: WP1
Target WP: WP1
Goals: The model must be able to represent actors handling assets.
Acceptance criteria: Relevant actors can access the data and assets they should be able to access.
Status: Agreed
Dependencies: None

Requirement R108

Requirement : Actors must be able to move in the location domain they belong to.
Source WP: WP1
Target WP: WP1
Goals: Domains reduce analysis artefacts by avoiding actors that enter computers, for example.
Acceptance criteria: For relevant actors in the case studies it is possible to specify a domain that they are bound to.
Status: Agreed
Dependencies: None

Requirement R109

Requirement : The assets associated with an actor must move with them.
Source WP: WP1
Target WP: WP1
Goals: Actors do not forget data or loose assets when moving.
Acceptance criteria: For relevant actors, the data and items they "know" is still available once they move.
Status: Agreed
Dependencies: None

Requirement R110

Requirement : Socio-technical security models must provide the necessary abstraction to represent policies.
Source WP: WP1
Target WP: WP1
Goals: Policies are an essential component of socio-technical systems.
Acceptance criteria: Relevant policies in the case studies can be represented in the model.
Status: Agreed
Dependencies: None

Requirement R111

Requirement : Policies must require a set of credentials and enable a set of actions, both possibly complete or empty.
Source WP: WP1
Target WP: WP1
Goals: To be able to decide whether a policy is fulfilled, we need to be able to check whether all credentials are provided, and which actions are enabled.
Acceptance criteria: For policies from the case studies, the enabling credential and the enabled actions can be represented.
Status: Agreed
Dependencies: None

Requirement R112

Requirement : Policies should be associated with data and/or locations.
Source WP: WP1
Target WP: WP1
Goals: Both data and locations can access control policies, so the policies must be stored there.
Acceptance criteria: Policies that are associated with data or locations can be stored in the model at the locations that represent them.
Status: Agreed
Dependencies: None

Requirement R113

Requirement : Socio-technical security models must support the storage of data for elements in the model.
Source WP: WP1
Target WP: WP1
Goals: Data represents properties of elements in the model, or analysis results. The model establishes the relation between elements and the data.
Acceptance criteria: For all elements in the model, arbitrary data can be stored at them and retreived.
Status: Agreed
Dependencies: None

Requirement R114

Requirement : The data must be retrievable and settable, based on the element, its type, or a certain action to be performed on it.
Source WP: WP1
Target WP: WP1
Goals: Some of the data is available from the beginning, other will be computed, so it must be accessible.
Acceptance criteria: There exist API functions to retrieve and set data based on certain attributes.
Status: Agreed
Dependencies: None

Requirement R115

Requirement : Socio-technical security models must support the storage of analysis results in the model.
Source WP: WP1
Target WP: WP1
Goals: Analysis results must be stored in the model either for elements or parts of the model.
Acceptance criteria: For all elements in the model, analysis data can be stored at them and retreived.
Status: Agreed
Dependencies: None

Requirement R116

Requirement : The model must support to associate model elements with attacks and vulnerabilities.
Source WP: WP1
Target WP: WP1
Goals: When considering changes to a part of the model, it must be clear which attacks this part is identified in.
Acceptance criteria: For all elements in the model, attacks and vulnerabilites can be stored at them and retreived.
Status: Agreed
Dependencies: None

Requirement R117

Requirement : The model must support locations that represent countermeasures.
Source WP: WP1
Target WP: WP1
Goals: The TREsPASS process not only identifies attacks, but also adds countermeasures. These must be storeable in the model.
Acceptance criteria: For relevant case studies, identified countermeasures can be represented in the model.
Status: Agreed
Dependencies: None

Requirement R118

Requirement : There should exist an API to the model that supports access to model elements and data associated with these elements.
Source WP: WP1
Target WP: WP1
Goals: When developing model tools, there must be an API to create the model and to access the data stored in the model.
Acceptance criteria: The TREsPASS tools access the model through the APIs, not directly.
Status: Agreed
Dependencies: None

Requirement R119

Requirement : Model elements must be accessible by their identifier.
Source WP: WP1
Target WP: WP1
Goals: Model parts can contain other parts, that are represented by their identifiers. The API must be able to obtain the model part with this identifier.
Acceptance criteria: The API provides functionality to obtain model elements.
Status: Agreed
Dependencies: None