predict
prioritise
prevent

TRESPASS

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

Deliverable D3.1.2

Final Requirements for Quantitative Analysis Tools

Project:	TRESPASS
Project Number:	ICT-318003
Deliverable:	D3.1.2
Title:	Final Requirements for Quantitative Analysis Tools
Version:	1.0
Confidentiality:	Public
Editor:	R. Kumar
Cont. Authors:	R. Kumar, Z. Aslanyan, M. Martins, R. Trujillo-Rasua, A. Lenin, J.C. van de Pol
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	Jaco van de Pol	Management Summary
UT	Rajesh Kumar	1, 2, 3, 4
DTU	Zara Aslanyan	3
UL	Rolando Trujillo-Rasua	3
CYB	Aleksandr Lenin	3
ITR	Miguel Martins	3

Quality assurance		
Role	Name	Date
Editor	Rajesh Kumar	2015-10-30
Reviewer	Miguel Martins	2015-10-15
Reviewer	Marlon Fraile Cestari	2015-10-15
Task leader	Jaco van de Pol	2015-10-30
WP leader	Jaco van de Pol	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

This document presents the evolution of the requirements during the project, focusing on the requirements related to WP3, *quantitative analysis tools*. We refined the initial requirements, brought up new requirements among WP3 and other work packages, and prioritised the requirements.

A project-wide *requirements task-force* is in place to guarantee a systematic approach to the refinement and consistency of the requirements. This deliverable therefore reflects the achievements of the task-force, projected to the requirements of WP3.

Key takeaways:

- WP3 is the computational analysis engine to the project. It acts as a bridge between on the one hand WP1 (construction of the socio-technical model) and WP2 (enrichment with quantitative data), and on the other hand WP4 (visualisation of analysis results) and WP6 (tool integration).
- Clearly, the quality of the analysis results crucially depend on the quality of the input models and data, imposing requirements on WP1 and WP2, but also on the input from the case studies, WP7 to validate the analysis models.
- WP3 also provides a bridge between the TRE$_S$PASS methodology to security of socio-technical systems, and the underlying mathematical models and algorithms. Consequently, WP3 analysis tools consist of different boxes, reflecting the underlying mathematical models, typically various forms of Markov models or Games.
- Together, these tool components realise the objective of the analysis tools to important features to be analysed in TRE$_S$PASS. In particular, we focus on the probability, time, effort, cost, and impact of attacks.
- Internally, model transformations enable the smooth cooperation between the various tool components. An User Interface abstracts from the underlying mathematics, and from the different data formats in use.

1. Introduction

WP3 serves as the computational engine of the TRESPASS project. Its goal is to develop state-of-the-art quantitative analysis tools that can identify attack scenarios, rank them and propose appropriate countermeasures to them. This is done by first deriving *attack DSL* from the socio-technical model (Socio-technical security model specification, (The TRESPASS Project, D1.2.2, 2015)) and then translating them into a lower level model such as MA (Markov automata), I/O IMC (Interactive Markov Chain), or TA (Timed automata) (The TRESPASS Project, D3.3.2, 2015) which are further analysed. A direct analysis route without going through step of attack DSL, thus from socio-technical model to qualitative and quantitative analysis models is also pursued in WP3. An overview of the different tools, techniques and metrics developed by the WP3 partners is provided in (The TRESPASS Project, D3.3.2, 2015).

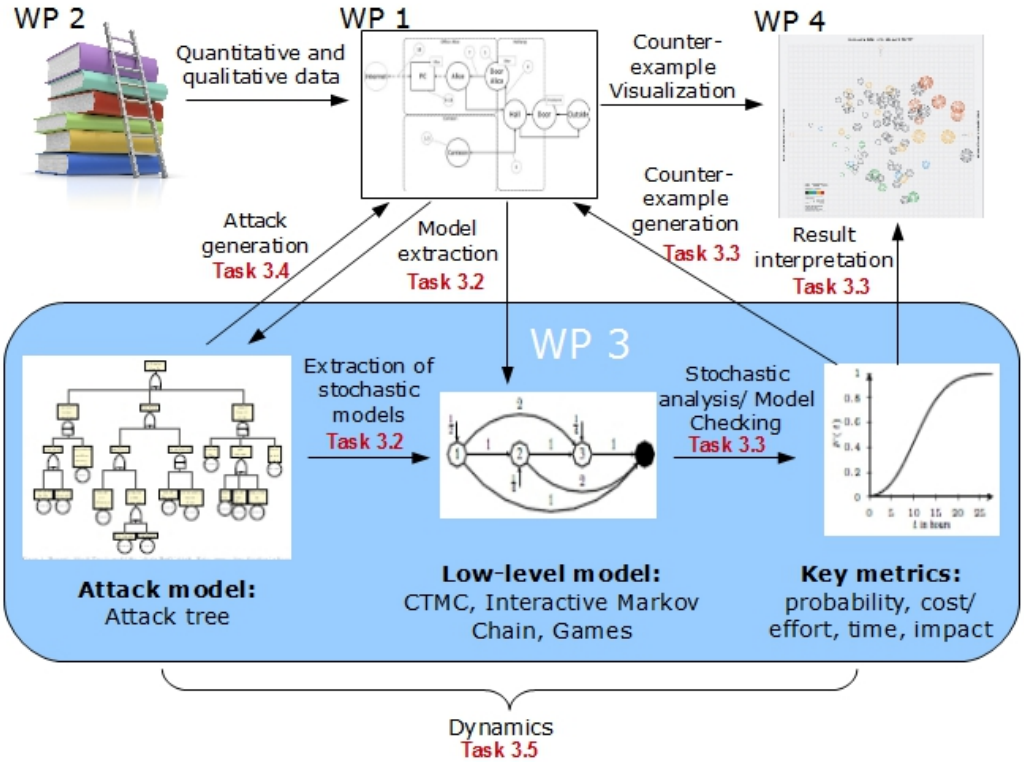

Figure 1.1.: The tasks and models of WP3 and their links with WP1, WP2 and WP4 within TRESPASS. An arrow from A to B indicates that A provides inputs to B.

Thus, WP3 forms an important interface between the socio-technical models developed in WP1 and visualization techniques developed in WP4. Quantitative data from WP2 (Data

management process) is gathered to support both tasks of model development in WP1 (Socio-technical security model specification) and annotating the atomic attack steps of attack DSL (as in WP3). The case studies provided by WP7 serve to validate the quality and applicability of the range of techniques developed in WP3. The set of algorithms and analysis techniques will be integrated into a single tool, to be included in the tool set produced by WP6. Figure 1.1 taken from (The TRE$_S$PASS Project, D3.1.1, 2013) summarizes the interaction of WPs in TRE$_S$PASS.

In order to proceed in a systematic manner, the entire WP3 process is split into various tasks. Task 3.1 is responsible for the gathering of requirements originating within the WP necessary for building tools and algorithms. The requirements addressed by other WPs (Work Packages) to WP3 are also elicited and critically examined, as a crucial part of this task.

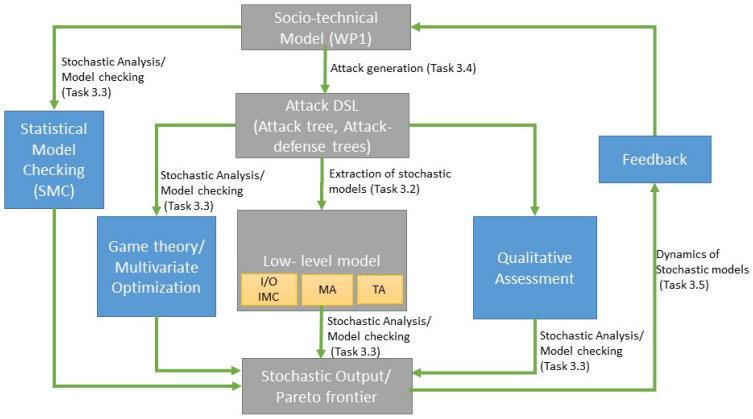

Figure 1.2.: Different formalisms being developed within TRE$_S$PASS. Here, I/O IMC refers to Input/Output interactive Markov chain, MA refers to Markov automata and TA refers to Timed automata, Attack DSL refers to attack domain specific language

Apart from task T3.1 which is the basis for the current deliverable, Task T3.4 extracts attack DSL from the WP1 socio-technical model, and defines its formal semantics. To keep an intermediate step of attack DSL is a well thought intention in WP3, and its choice is favourable geared towards its expressiveness and intuitive representations. Task T3.2 extracts more abstract mathematical structures (lower level formalism), such as labelled transition systems from attack DSL. This is useful to represent dynamic behavioural properties such as causal and temporal dependencies.

Task T3.3 uses either attack DSL or lower level formalism and is concerned with developing several methodologies and algorithms to quantify risks. Thus the prime focus of this task is to answer questions such as, “What is the probability of attack within t days?”,

"What is the cost optimal path for an attacker given his skills and resources (budget and time constraints)" and "Which countermeasures should a risk manager prioritize in his enterprise over the short term" Task T3.5 will investigate how the techniques and algorithms developed can be parametrized to handle the dynamics in the socio-technical model efficiently. Figure 1.2 provides a succinct representation of these tasks.

1.1. Goals

This document provides the set of final requirements for WP3 as evolved in the TRE$_S$PASS project as part of task T3.1. Its objective is to position WP3 in line with the TRE$_S$PASS general workflow (The TRE$_S$PASS Project, D6.2.2, 2015) and identify a set of dependencies which need to be conformed from WP3 perspective in order to provide a coherence between the other WPs and different analysis tools developed as part of this WP. More specifically, this deliverable revises and extends the set of defined requirements based on research and case analyses performed since month 6 (The TRE$_S$PASS Project, D3.1.1, 2013) to month 30 of the project.

We specifically look into:

- Identifying the functional and use case requirements from (The TRE$_S$PASS Project, D6.2.2, 2015) relevant to WP3.
- Analysing the feasibility and milestones of requirements that are directed to WP3 from other WPs;
- Emphasising the requirements relevant to multiple WPs;
- Developing WP3 analysis techniques and tools to be integrated by WP6;

1.2. Document structure

Appendix B provides the context for this deliverable in the TRE$_S$PASS project. It describes the overall summary of the project and the TRE$_S$PASS workflow. It provides an overview of conceptual system architecture, supported components and stages: data collection, model creation, analysis and visualisation which forms the basis of development of TRE$_S$-PASS workflow.

In chapter 2 we first provide a general overview of the TRE$_S$PASS workflow (The TRE$_S$-PASS Project, D6.2.2, 2015). This provides a clarified scope, context and position of WP3 from the functional perspective of the TRE$_S$PASS tool-kit. In chapter 3 we provide a brief description of state of art analysis tools developed in WP3 and elicit their functional requirements.

In chapter 4, we provide a functional decomposition of the WP3 goal used to generate a set of requirements and commitments that are the constraints, assumptions and dependencies of WPs for accomplishing the overall goals of TRE$_S$PASS.

Lastly, in chapter 5, we provide a subset of the requirements that we deem most important and need our focus during months M36-48. Appendix A provides a succinct representation in form of requirement table that specify requirements originating from WP3 to other WP and the commitments of WP3 to be accomplished in a time bounded manner.

1.2.1. Foreground and Background

This deliverable aims to consolidate and realistically assess the requirements dealt with in the project, either in previous deliverables or in review meetings. This final deliverable follows the (external) deliverable (The TRE$_S$PASS Project, D3.1.1, 2013) on "Initial Requirements for Quantitative Analysis Tools" and internal deliverable (The TRE$_S$PASS Project, I3.1.1, 2014) on "Immediate Requirements for Quantitative Analysis Tools". It logically extends from the lessons learnt and presented in other deliverables (mainly in (The TRE$_S$PASS Project, D1.3.1, 2013)) and WP3 (mainly in (The TRE$_S$PASS Project, D3.1.1, 2013) and (The TRE$_S$PASS Project, D3.4.1, 2014)).

As a foreground, we enlist the updated status and provide the WP3 vision on requirements directed to WP3 in chapter 4. Here, we also describe the requirements raised by other WPs for WP3 to fulfill. As refinement of requirements is an iterative process, this document reports the current status of dependencies in TRE$_S$PASS which evolved during the course of project and its contents are almost 50 % new.

2. Positioning of WP3 in the TRESPASS workflow

In this chapter, we look at the TRESPASS work flow as developed in (The TRESPASS Project, D6.2.2, 2015) (presented in figure 2.1) to highlight the functional coupling of WP3 analysis tasks and figure 2.2 to understand the magnified operational details. The analysis stage specifically relies on several processes and databases which are being built in TRESPASS. Some of these process are manual while others are automated. The algorithms and analysis runs in a separate tool which remains hidden from user but the output will be accessible at the user-interface.

2.1. TRESPASS Integration Diagram

WP3 as in figure 2.2, takes several inputs: attack navigator maps, attacker profiles, attack pattern library and produce output results which can be used for scenario analysis, risk analysis and root cause analysis.

Inputs required for analysis : WP3 takes attack scenarios as input for running the analysis. These attack scenarios are constructed using a description of model, expert knowledge base gathered at data collection stage and a user-defined interface. The prime processes involved are:

Attack navigator maps : An Attack navigator map is a graphical diagram based on a model of system or organization through which all attack scenarios, relevant to asset can be generated. Building an attack navigator map involves a model pattern library involving standard TRESPASS components (as developed in WP1/WP5).

Attacker profile : An attacker profile includes a description of adversary along with his motivation/goals, strategy, capability,resources, knowledge of system and initial access. For the TRESPASS model analysis, it is envisaged that there will be an attack profile library. The responsible WP for development are WP2/WP5/WP6.

Attacker pattern library : An attack pattern library is a knowledge base for further refinement of attack steps learnt through empirical studies. This is being developed by WP2/WP5.

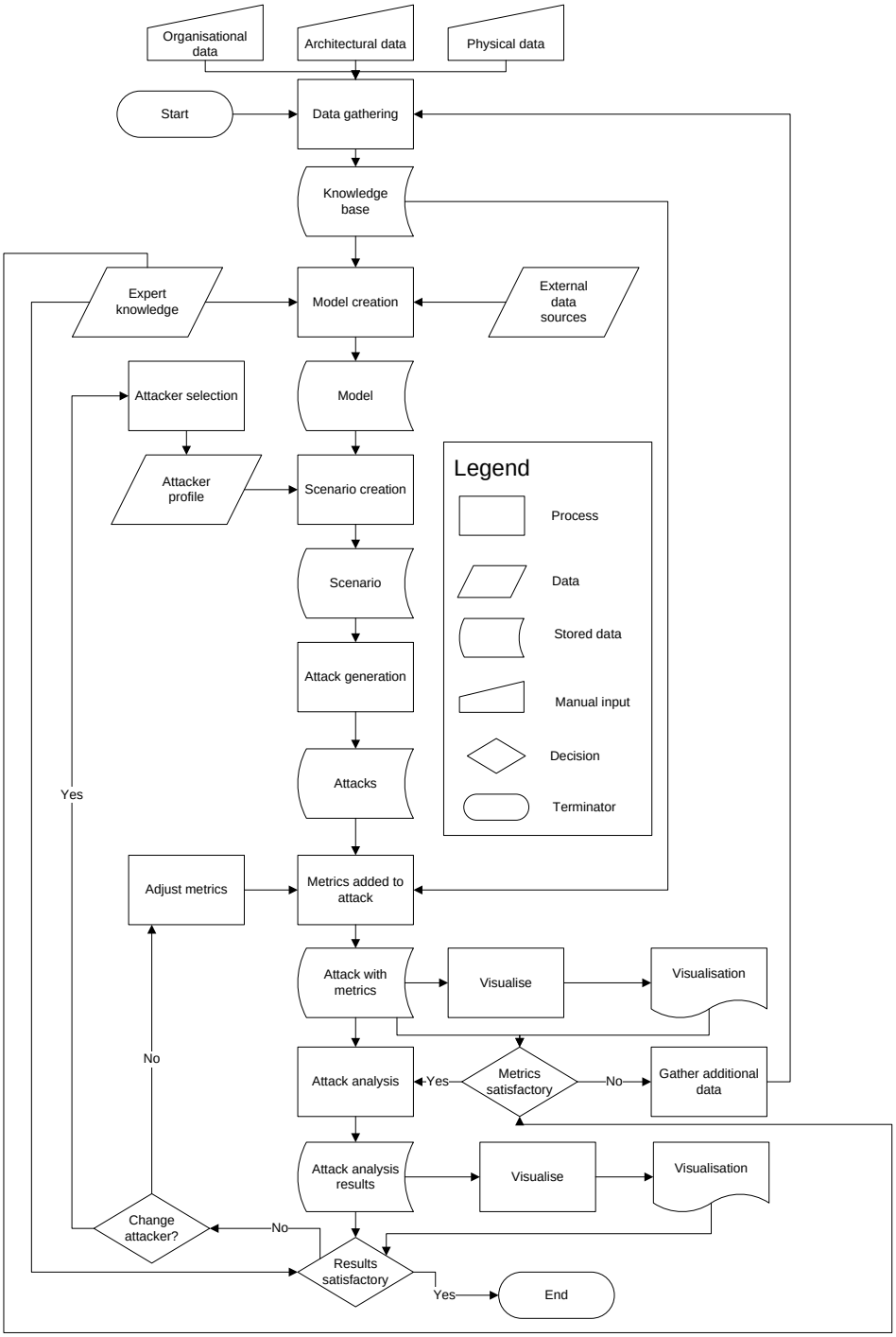

Figure 2.1.: The tasks and models of WP3 and their links with other WPs within TRESPASS.

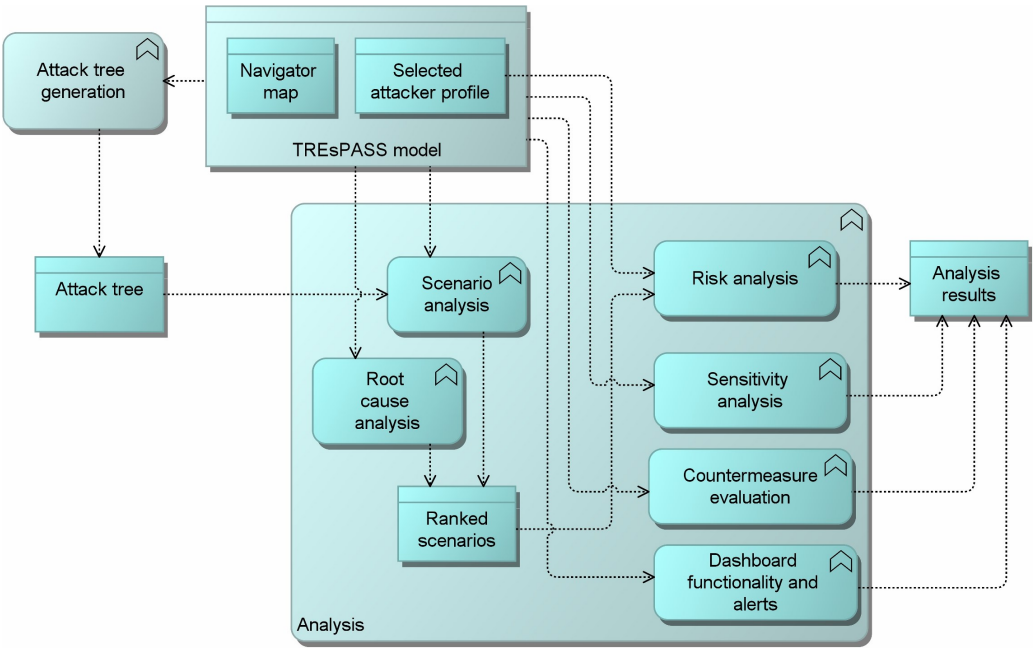

Figure 2.2.: Functional desciption of WP3 within TRESPASS.

Data management and retrieval : WP2 is responsible for gathering physical, social and digital data that will be used to build the model and also annotate the basic attack steps in the attack tree.

Output of the analysis : The outcome of the analysis tools will be primarily used for visualization by WP4 and also provided to WP1 to update the model.

Functional description of analysis modules : The several tools being developed by partners in WP3 enables .

Scenario analysis : WP3 is responsible of extracting a behavioural model from the attack scenario which is used to quantify the system vulnerabilties. Generally, the analysis provides the user with a set of attack values (Static and time dynamic probability to reach goal, Optimal Cost, Optimal time) and associated attack path in reaching the goal. A trade-off between the different attack values is also depicted, as Pareto frontier which can answer the questions on attacker utility (Expected payoff vs Cost/Time to execute the attack).

Risk analysis : Given, an estimate of frequency of attack scenarios and probable loss magnitude to the organization, WP3 can quantify the risk associated with scenarios as annual loss expectancy.

Root cause analysis : Users of the TRESPASS model analysis might be interested in the most vulnerable attack steps that lead to system compromise. Hence, WP3 provides the user a flexibility to choose the attack steps and obtain quantified attack scenarios which are then also ranked according to an attack metric (Optimum cost/Optimum time, Minimum penalty in reaching the goal).

Senstivity analysis : In order to account for inaccuracies in input values, WP3 provides a sensitivity analysis.

Countermeasure evaluation : WP3 provides a cost/benefit analysis of potential countermeasures by assessing its impact in terms of attack values. It is helpful for risk managers to target their investments and working under limited budgets.

Use case requirements : Though the functional requirements are important, use case requirements serves as bechmark for the development of tools.

User friendliness : All the tools integrated under one umbrella and tightly bound with other WP must have a common interface. This will help user obtain the specific information which he is looking for without having a deep understanding of underlying data types and algorithms. The interface should be friendly enough to let him know about the potential choices he has. Example : One can obtain static probabilities, cost required to reach the goal or he can also obtain evolution of attack, given the attack cost , skill and efforts to successfully complete basic steps.

Results interpretation The results obtained by analysis should relate with the socio-technical model in the first place and should be concise and specific so that a user understands it. This will also be useful then to share it with WP4 for further visulization.

Modularity The analysis tools and techniques should be modular to provide user flexibility to adapt the model according to his specific enterprise needs. WP3 should envisage of having compositional models which are easy to understand, modify and extend.

3. Description of analysis tools developed in WP3

WP3 develops multiple analytical tools each having its own peculiarities in quantifying risk. This is certainly an advantage in the project but it comes with a additional constraint on requirements for a common data format and structure which every tool can understand. Hence, we briefly discuss each tool and also elicit its specific requirements.

3.1. ATCALC

3.1.1. Relevance of Tool

ATCalc (Arnold, Belinfante, van der Berg, & Stoelinga, 2013), extends classical attack trees with a notion of time; inspired by the fact that there is a strong correlation between the amount of resources in which the attacker invests (in this case time) and probability that an attacker succeeds. It uses stochastic model checking (SMC) and compositional aggregation as an engine to compute the evolution of attack. Morever, it also takes into account the dependencies between basic attack steps and can also evaluate shared subtrees.

3.1.2. Technical characteristics

ATCALC can be used by downloading a stand-alone version accessible via git from `http://fmt.cs.utwente.nl/tools/scm/dftcalc.git` in the branch `atcalc`, and via a web interface accessible at `http://fmt.ewi.utwente.nl/puptol/atcalc/`. ATCALC is open source, but requires a license for CADP, which is free for academic institutions. The web interface is realized with the use of PUPTOL (Belinfante & Rensink, 2013) and extends the downloadable version with a GUI as well as the plot function. It allows the user to

1. input AT(Attack tree) models via a text screen;
2. select the dependability metrics. This can be
   - the probability of an attack for several attack times t, or
   - the probability on an attack during an interval $[T1, T2]$, or
   - the mean time for a successful attack;

3. set various options: which model checker to use; the error bound, the level of verbosity, and whether to color output;
4. choose to store and plot a graph.

The results can be given either by numbers, via the button *show result*, or as a plot, via the button *plot result*. The input and configuration of the web interface can be saved via the button *permalink.*

3.1.3. Open issues

ATCALC will be integrated in the TRE$_S$PASS platform developed by WP6. It enables time-dynamic analysis, but even more important is its flexibility: Each type of behavior can be quickly encoded as gates or leaves and added to the framework. Specific issues pertaining to integration are :

1. CADP used for composing and minimizing IMCs is not an open source software-though we future we plan to integrate with LTSmin

3.1.4. Requirements of tool from other WP

1. WP2: ATCALC takes annotated attack tree as the input. Thus, the atomic steps of attack tree requires information about the attack execution time and probability of success. This specific information needs to be provided by WP2, responsible for data management.
2. WP1, WP3– To integrate ATCALC in a broad project perspective, we need to have a clear understanding of data formats (though it is recognised that it will be a variant of .xml format)

3.2. ATEvaluator

3.2.1. Relevance of Tool

Attack trees are a widely used graphical tool for modelling the security threats of an organisation and representing attack scenarios in an intuitive manner. The root of a tree represents the main goal of an attacker, and the leaves correspond to an attacker's basic actions. Standard attack trees combine basic actions either conjunctively or disjunctively, thereby limiting their expressiveness. Most analyses of attack trees consider attack tree with one parameter and optimise one aspect of an attack scenario, such as feasibility or cost of an attack. Moreover, in most attack tree models with multiple parameters values, characterising basic attacks, are propagating to the root based on the local decision strategies. In case of incomparable values, this approach may yield sup-optimal results.

Our approach To overcome these limitations, we present evaluation techniques that consider attack trees with multiple parameters. Our evaluate techniques optimise all parameters at once. We handle conflicting parameters by computing the set of optimal solutions in terms of Pareto efficiency. A solution is called Pareto efficient if it is not dominated by any other solution (Legriel, Guernic, Cotton, & Maler, 2010).

3.2.2. Technical characteristics

The evaluation technique considers basic actions characterised by more than one parameter (e.g. probability *and* cost) for analysing complex scenarios. It computes different aspects of an attack scenario and handles multiple objectives. Furthermore, as different objectives may conflict with each other, the technique considers the set of Pareto optimal solutions to handle the analysis of incomparable values.

We consider attack trees with negations, similar to attack-defence trees. Depending on whether actions only occur positively or negatively in an attack tree, we can make certain assumption w.r.t. the tree's evaluation.

We evaluate attack trees in the Boolean and probabilistic settings, and annotate actions with only one kind of cost or multiple cost, where we apply Pareto techniques.

3.2.3. Open issues

We need to understand better the effect of reliability of available data on the reliability of computed results, and how to account for different rankings between multiple costs. Dealing with more than 2 or 3 costs is technically not an issue, but needs special support for explaining computed solutions.

3.2.4. Requirements of tool from other WP

1. WP2: The evaluation of an attack tree is done by assigning values to the basic actions. Thus, the evaluation technique requires quantitative measures on the basic actions of an attack tree, such as the probability of success and the cost of an attack.
2. WP1, WP3- The evaluation technique requires an attack tree model. Thus, we need to have a clear understanding of data formats.

3.3. Attack Tree Analyzer

3.3.1. Relevance of Tool

The tool performs quantitative attack tree analysis. The type of analysis and the outcome depends on the chosen model. Currently the two models are supported: the failure-free model (Buldas & Lenin, 2013) and the parallel model (Lenin, Willemson, & Sari, 2014). If the failure-free analysis is launched, the outcome is a binary value which gives an answer to the question whether the considered infrastructure is a fruitful target for rational profit-oriented attackers. If the system is analyzed by the parallel model, the result is the most profitable attack vector (if any). The analysis can be done taking attacker profiles into account, as well as without profiling considerations.

3.3.2. Technical characteristics

The tool expects to receive an XML-encoded attack tree. For the parallel model two approximation algorithms are available: the genetic algorithm and the adaptive genetic algorithm. For the failure-free model 4 algorithms are available: the precise algorithm, the propagation algorithm, the genetic as well as adaptive genetic algorithms. The result is an XML file. If the result is a profitable attack vector (as in the case of analysis by the parallel model), the output XML file contains the entire subtree corresponding to the most profitable vector. In the case of the failure-free analysis the result is the attacker utility value. An output XML file may as well contain status and error messages. If attacker profiling is desired, attacker profiles need to be specified.

3.3.3. Open issues

Subsequent work will concentrate on increasing the algorithm performance, and on analysing attack graphs containing fan-ins.

3.3.4. Requirements of tool from other WP

1. WP1: Each node in the generated attack tree must contain no less than two children.
2. WP1, WP2, WP5: Provide meaningful attack tree which is ready for analysis.
3. WP1, WP2: The tool supports attacker profiling, thus it is required that the attacker profiles are made available prior to analysis.

3.4. The ADTool

The ADTool is aimed at providing security consultants and academic researchers with a rigorous and user-friendly application that supports security analysis based on attack–defense trees. From a formal perspective, attack trees, protection trees, and defense trees are instances of attack–defense trees. Therefore, the ADTool can also be employed to automate and facilitate the usage of all aforementioned formalisms.

ADTrees in the ADTool are evaluated in a bottom-up fashion. The operators used during the bottom-up computation differ depending on the type of attribute considered. Attribute types supported by the ADTool are: attributes based on real values (e.g., time, cost, probability), attributes based on levels (e.g., required skill level), and Boolean properties (e.g., satisfiability of a scenario). All these attributes can be synthesized from the point of view of an attacker (e.g., the cost of an attack), of a defender (e.g., the cost of defending a system), or of both (e.g., overall maximum power consumption).

The ADTool requires the user to populate the relevant non-refined nodes directly on the tree or uses an overview table. The use of the table is particularly helpful in case of large models. The tool ensures that the provided values are consistent and belong to a specified value domain.

3.4.1. Requirements of the ADTool from other WPs

1. WP1-WP3: Attack–defense trees are expected to be automatically created from system models. Consequently, those generation tools should be able to export the generated ADTrees complying with the specification of the ADTool's XML scheme.
2. WP2: The evaluation of an attack–defense tree is performed by assigning values to non-refined nodes. Quantitative values on these basic actions can be provided either through the tool GUI or by importing them from an XML file. Starting from Version 1.3, the ADTool allows users to import/export XML files containing information related to attribute values stored at the nodes of ADTrees.
3. WP6: the ADTool is expected to be integrated in the TRE$_S$PASS platform developed by WP6. Perhaps, the best communication means between the ADTool and other similar tools is the XML Scheme specified in the ADTool manual. The development of source-to-source compilers between the different XML Schemes of the considered tools is thus a need.

4. Description of requirements

To enable systematic discussion, elicitation and cohesion among the different Work Packages, a set of *derived requirements* listed as **Requirements table** (also refer Table A), has been developed. This provides a succinct view of the dependencies promoting collaborative development process. Further details of the requirement table can be found in (The TRESPASS Project, D6.2.2, 2015).

Here, we put up a set of requirements from the requirement table of which WP3 is the source – that is the WP that want the requirements to be implemented with a acceptance criterion to judge its fulfilment. We also provide a list of requirements directed to WP3 by other WPs addressing their concerns in a time bound way.

A set of requirement as:

Identifier : Unique identifier id taken from requirement table;

Requirement : Specification of requirement

Source WP: Source WP is the originator of the requirement which deem the requirement as functionally relevant to fulfil its own tasks.

Target WP: Target WP is the WP that first accepts the requirement, evaluates it and then mutually agree with source WP to fulfil the requirement in a time bound manner.

Goals: They serve as justification of the requirement from the origin WP and the basis for target WP to understand and respond it.

Acceptance criteria: Source WP sets certain criterion over which a raised requirement by it needs to be fulfilled and also serves as benchmark for target WP to complete it.

Status: This refers to the current position of the requirement if it has been already completed or is in process of acted on, in accordance to the acceptance criterion of the source WP

Dependencies: It refers to the cross links that the requirement imposes. To be precise, on its completion, which other requirements or tasks are progressed.

4.1. Requirements directed by WP3

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 R37

Requirement : Identification of required analysis measures and quantities

Source WP: WP3

Target WP: WP5

Goals: In order to develop dedicated analysis methods, we need a clearer idea of the kind of properties to be analysed. This is needed for task 3.3

Acceptance criteria: Agreed-upon list of measures/quantities

Status: Completed

Dependencies: Task 3.3 depends on this.

Requirement R38

Requirement : Attack tree generation from socio-technical model

Source WP: WP3

Target WP: WP3

Goals: This is an indispensible part of the intended tool chain. This is needed for task 3.4

Acceptance criteria: Working tool

Status: Agreed

Dependencies: Task 3.4 depends on this.

Requirement R39

Requirement : Realistic data with regard to structure, frequency and costs of common attacks for each case study.

Source WP: WP3

Target WP: WP7

Goals: For realistic output, we need realistic input

Acceptance criteria: Data that is as realistic as possible (given the constraints of company-confidentiality), in a format that can be processed by the tools'

Status: Agreed

Dependencies: None

Requirement R40

Requirement : Support for attack tree visualisation

Source WP: WP3

Target WP: WP4,6

Goals: Improved presentation of tool output

Acceptance criteria: Visual inspection

Status: Agreed

Dependencies: None

Requirement R42

Requirement : Data analysis model relevant to case studies

Source WP: WP3

Target WP: WP4

Goals: Require capabilities to sufficiently analyse data for case study (WP7) purposes

Acceptance criteria: A means to visualise the analysis results from WP3 tools, understood by consultants"

Status: Agreed

Dependencies: Agreed subject to data being passed in a timely manner from WP2 and analysis capability clearly defined by WP3. This is also agreed subject to agreement being reached as to the visualisation that can be developed in the time available

4.2. Commitment of WP3

Requirement R07

Requirement : Ability to generate an attack tree

Source WP: WP5

Target WP: WP3

Goals: Needed to proceed with Task 5.3

Acceptance criteria: The tool exists and runs correctly

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 R23

Requirement : The TREsPASS approach needs to account for (partial) risks that are induced through the properties of the entities (products and services) involved.

Source WP: WP7

Target WP: WP3

Goals: Needed in order to incorporate telco scenarios

Acceptance criteria: For sufficiently large models the analysis should be faster than manual analysis.

Status: Agreed

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 R25

Requirement : The TREsPASS approach should handle complex correlations of partial risks

Source WP: WP7

Target WP: WP3

Goals: Needed for case studies

Acceptance criteria: Should work when tested with telco scenarios

Status: Shelved

Dependencies: Reason for shelving: we don?t know how to do this mathematically, so we will not be able to implement this in the an alysis tools.

Requirement R26

Requirement : It should be possible to generate attack scenarios from the model.

Source WP: WP7

Target WP: WP3,5

Goals: Needed for case studies

Acceptance criteria: Generation of attack scenarios from the telco models

Status: Agreed

Dependencies: R26

Requirement R27

Requirement : The TREsPASS tool should provide solutions or mitigation strategies for attack scenarios.

Source WP: WP7

Target WP: WP3,5

Goals: Needed for case studies

Acceptance criteria: Mitigation strategy for the identified attack scenario

Status: Shelved

Dependencies: Reason for shelving: in line with the discussion that occurred in WP5 and the points raised by the advisory board that the scope of our work is not impact analysis.

Requirement R31

Requirement : The analysis tool should provide for the possibility to generate attack scenarios based on the structural components described in the requirements to WP1.

Source WP: WP7

Target WP: WP3

Goals: Needed for case studies

Acceptance criteria: Identical to R24: generation of attack scenarios from the models

Status: Agreed

Dependencies: None

Requirement R32

Requirement : The analysis tool should stipulate the provisioning of solutions or mitigation strategies for attack scenarios.

Source WP: WP7

Target WP: WP3

Goals: Needed for case studies

Acceptance criteria: Identical to R24: 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: Shelved

Dependencies: Reason for shelving: in line with the discussion that occurred in WP5, that the scope of our work is not impact analysis.

Requirement R38

Requirement : Attack tree generation from socio-technical model

Source WP: WP3

Target WP: WP3

Goals: This is an indispensible part of the intended tool chain. This is needed for task 3.4

Acceptance criteria: Working tool

Status: Agreed

Dependencies: Task 3.4 depends on this.

Requirement R56

Requirement : The system can suggest updates to the TRESPASS model based on scenarios and associated parameters.

Source WP: MT

Target WP: WP3,5,6

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 1 (Security Investment), Use-case 2 (Audit) and Use-case 5 (Quick scan), specifically U1.12, U1.13, U2.4 U5.6.

Acceptance criteria: 80 % of the users specified in the Use Cases are able to respond effectively to the suggested updates

Status: Shelved

Dependencies: Reason for shelving: adaptations to the sociotechnical model will have to be done manually, not automagically.

Requirement R70

Requirement : The system should be able to generate an attack tree from a map, an attacker profile, a target asset, and an attacker position.

Source WP: MT

Target WP: WP3

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U2.8, U4.4 and U5.10.

Acceptance criteria: All paths in the generated attack tree correspond to rational at- tack paths according to map definitions and attacker utility. All rational paths appear in the attack tree. Quantitative values of actions correspond to the values on the map.

Status: Agreed

Dependencies: None

Requirement R71

Requirement : The system should be able to generate an attack tree from a map, an attacker profile, and an attacker position, based on attacker utility.

Source WP: MT

Target WP: WP3

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U2.8, U4.4 and U5.10

Acceptance criteria: All paths in the generated attack tree correspond to rational at- tack paths according to map definitions and attacker utility. All rational paths appear in the attack tree. Quantitative values of actions in the tree correspond to the values on the map.

Status: Agreed

Dependencies: None

Requirement R72

Requirement : The user should be able to run scenario analysis on a navigator map with associated attacker profile.

Source WP: MT

Target WP: WP3

Goals: Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 4 (Product-service system), specifically U4.4.

Acceptance criteria: 80 % of the users specified in the Use Cases are able to use the interface effectively after receiving the specified training

Status: Agreed

Dependencies: None

Requirement R73

Requirement : The user should be able to run scenario analysis on an attack tree with associated attacker profile.

Source WP: MT

Target WP: WP3

Goals: Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation).

Acceptance criteria: 80 %. of the users specified in the Use Cases are able to use the interface effectively after receiving the specified training.

Status: Agreed

Dependencies: None

Requirement R74

Requirement : The system should present at most 7 scenarios to the user at once.

Source WP: MT

Target WP: WP3

Goals: Supports Use-case 5 (Quick scan), specifically U5.1

Acceptance criteria: 80 %. of the users specified in the Use Cases indicate that they can understand the default view of ranked scenarios.

Status: Agreed

Dependencies: None

Requirement R75

Requirement : The tools can calculate the total risk associated with a ranked set of scenarios and a set of attacker profiles.

Source WP: MT

Target WP: WP3,5

Goals: Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 1 (Security Investment), Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U1.11, U2.9, U4.4 and U5.11.

Acceptance criteria: The output is consistent with the mathematical definition.

Status: Completed

Dependencies: None

Requirement R76

Requirement : The output is consistent with the mathematical definition.

Source WP: MT

Target WP: WP3

Goals: Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 1 (Security Investment), Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U1.11, U2.9, U4.4 and U5.11.

Acceptance criteria: The output is consistent with the mathematical definition.

Status: Agreed

Dependencies: None

Requirement R77

Requirement : The system should enable an analysis of the most likely scenarios that led to compromise of a particular asset

Source WP: MT

Target WP: WP3

Goals: Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 1 (Security Investment), Use-case 2 (Audit) and Use-case 5 (Quick scan), specifically U1.11, U1.17, U2.9 and U5.11.

Acceptance criteria: The output is consistent with the mathematical definition.

Status: Agreed

Dependencies: None

Requirement R78

Requirement : The TRESPASS tools should be able to calculate the effect of differences in the model on the risk values

Source WP: MT

Target WP: WP3

Goals: Derived from P5 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 1 (Security Investment) and Use-case 4 (Product-service system), specifically U1.2 and U4.4

Acceptance criteria: The output is consistent with the mathematical definition.

Status: Agreed

Dependencies: R05

Requirement R79

Requirement : The TRESPASS tools can calculate the effectiveness of a countermeasure in a navigator map from the map, a set of attacker profiles, and a cost function of the countermeasure.

Source WP: MT

Target WP: WP3

Goals: Derived from P6 (TRESPASS tools should be able to calculate the cost-effectiveness of a proposed countermeasure). Supports Use-case 1 (Security Investment), Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U1.2 U2.11, U4.2 and U5.13

Acceptance criteria: The output is consistent with the mathematical definition.

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: 80% of the users specified in the Use Cases are satisfied with the suggested countermeasures. 80% of users are able to successfully include countermeasures them into the model.

Status: Shelved

Dependencies: None

5. Conclusion

This deliverable focused on gathering and systematising the requirements for TRE$_S$PASS framework and workflow. Owing to the maturity and ongoing work of integration of the project, currently there are requirements that span more than one Work Package. We enlist here the most important of these from the perspective of WP3.

H3.1. *Attack tree generation tool.* This is one of the most important requirements directed from other WPs to WP3. Up till now most attack trees have been drawn manually by taking attack scenarios illustrated in case studies (developed in WP-7). However, currently it is also possible to generate attack trees from the socio-technical model automatically, using TreeMaker.

WPs involved primarily: WP3, WP1

H3.2. *Obtaining realistic data with the required parameters.* The analysis results obtained by WP3 asserts its meaning only if provided with realistic input models and data. Even though there has been quite some progress in understanding the process of gathering the data, it is clear that obtaining case specific data is hard. The case studies carried out in the TRE$_S$PASS-project provided the insight that many attack steps are repeated. This gave a clue for filling an attack pattern library. It is still to be established what is the right granularity to decorate the attack steps satisfactorily with feasible data.

WPs involved primarily: WP2, WP7

H3.3. *The analysis tool should provide analysis results based on the structural components defined in the socio-technical model.* This requirement involves several WPs, where a user can build his attack scenarios based on the model pattern library and structural components defined in socio-technical model (WP1) through a user interface (WP6).

WPs involved primarily: WP1, WP2, WP3, WP5, WP4, WP6.

H3.4. *Countermeasures in socio-technical model.* The framework of risk analysis in TRE$_S$PASS provides the user an enumerated list of possible attacks. Within this setting, WP3 also focusses on providing effective countermeasures to the considered attack scenarios, which could be traced back to the model itself. Since the interpretation of counter-measures is subjective, further case studies need to clarify the possible role of WP3 beyond evaluating the model after modification with the countermeasures.

WPs involved primarily: WP7.

H3.5. *Ensuring scalability.* Scalability is an issue that touches upon every link in the TRESPASS tool chain; in the context of WP3, it mostly concerns the formal complexity of the analysis methods. One of the ways to achieve scalability is through compositionality.

WPs involved primarily: WP1, WP2, WP6.

References

Arnold, F., Belinfante, A., van der Berg, F., & Stoelinga, M. (2013). DFTCalc: A tool for efficient fault tree analysis. In *32nd international conference on computer safety, reliability and security (safecomp'13)* (pp. 293–301). Springer.

Belinfante, I. A., & Rensink, D. A. (2013, June). *Publishing your prototype tool on the web: Puptol, a framework* (No. TR-CTIT-13-15). Enschede, the Netherlands: University of Twente, Centre for Telematica and Information Technology (CTIT). Retrieved from http://doc.utwente.nl/86255/

Buldas, A., & Lenin, A. (2013). New efficient utility upper bounds for the fully adaptive model of attack trees. In S. K. Das, C. Nita-Rotaru, & M. Kantarcioglu (Eds.), *Decision and game theory for security - 4th international conference, gamesec 2013, fort worth, tx, usa, november 11-12, 2013. proceedings* (Vol. 8252, pp. 192–205). Springer. Retrieved from http://dx.doi.org/10.1007/978-3-319-02786-9_12 doi: 10.1007/978-3-319-02786-9_12

Legriel, J., Guernic, C. L., Cotton, S., & Maler, O. (2010). Approximating the pareto front of multi-criteria optimization problems. In *Tacas* (p. 69-83).

Lenin, A., Willemson, J., & Sari, D. P. (2014). Attacker profiling in quantitative security assessment based on attack trees. In *Secure IT Systems, 19th Nordic Conference, NordSec 2014* (Vol. 8988, pp. 199–212). Springer.

The TRE$_S$PASS Project, D1.2.2. (2015). *Final policy-specification language.* (Deliverable D1.2.2)

The TRE$_S$PASS Project, D1.3.1. (2013). *Initial prototype of the socio-technical security model.* (Deliverable D1.3.1)

The TRE$_S$PASS Project, D3.1.1. (2013). *Initial requirements for quantitative analysis tools.* (Deliverable D3.1.1)

The TRE$_S$PASS Project, D3.3.2. (2015). *TRE$_S$PASS methods for stochastic analysis.* (Deliverable D3.3.2)

The TRE$_S$PASS Project, D3.4.1. (2014). *Attack generation from socio-technical security models.* (Deliverable D3.4.1)

The TRE$_S$PASS Project, D6.2.2. (2015). *Final refinement of functional requirements.* (Deliverable D6.2.2)

The TRE$_S$PASS Project, I3.1.1. (2014). *Intermediary requirements for quantitative analysis tools.* (Internal Deliverable I3.1.1)

A. Requirement table from WP3 perspective

Table A.1.: What WP3 requires from other WPs

#	requirement	source	target	goals	priority
R11	There must exist a clear description of the models.	WP3	WP1	Needed to produce visualisation toolkit	5
R37	Identification of required analysis measures and quantities	WP3	WP5	In order to develop dedicated analysis methods, we need a clearer idea of the kind of properties to be analysed. This is needed for task 3.3	
R39	Realistic data with regard to structure, frequency and costs of common attacks for each case study.	WP3	WP7	For realistic output, we need realistic input	
R40	Support for attack tree visualisation	WP3	WP4,6	Improved presentation of tool output	
R42	Data analysis model relevant to case studies	WP3	WP4	Require capabilities to sufficiently analyse data for case study (WP7) purposes	

Table A.2.: What other WPs require from WP3

#	requirement	source	target	goals	priority
R07	Ability to generate an attack tree	WP5	WP3	Needed to proceed with Task 5.3	5
R17	Map analysis results back to model	WP1	WP3	Needed to communicate analysis result back to TREsPASS tools	3
R23	The TREsPASS approach needs to account for (partial) risks that are induced through the properties of the entities (products and services) involved.	WP7	WP3	Needed in order to incorporate telco scenarios	
R24	Risks and relations should include contractional agreements and jurisdictional requirements.	WP7	WP1,3	Needed in order to incorporate telco scenarios	3
R26	It should be possible to generate attack scenarios from the model.	WP7	WP3,5	Needed for case studies	
R31	The analysis tool should provide for the possibility to generate attack scenarios based on the structural components described in the requirements to WP1.	WP7	WP3	Needed for case studies	
R70	The system should be able to generate an attack tree from a map, an attacker profile, a target asset, and an attacker position.	MT	WP3	"Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U2.8, U4.4 and U5.10."	
R71	The system should be able to generate an attack tree from a map, an attacker profile, and an attacker position, based on attacker utility.	MT	WP3	"Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U2.8, U4.4 and U5.10"	
R72	The user should be able to run scenario analysis on a navigator map with associated attacker profile.	MT	WP3	"Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 4 (Product-service system), specifically U4.4."	
R73	The user should be able to run scenario analysis on an attack tree with associated attacker profile.	MT	WP3	Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation).	

Table A.2.: What other WPs require from WP3

#	requirement	source	target	goals	priority
R74	The system should present at most 7 scenarios to the user at once.	MT	WP3	Supports Use-case 5 (Quick scan), specifically U5.1	
R75	The tools can calculate the total risk associated with a ranked set of scenarios and a set of attacker profiles.	MT	WP3,5	"Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 1 (Security Investment), Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U1.11, U2.9, U4.4 and U5.11."	
R76	The output is consistent with the mathematical definition.	MT	WP3	"Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 1 (Security Investment), Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U1.11, U2.9, U4.4 and U5.11."	
R77	The system should enable an analysis of the most likely scenarios that led to compromise of a particular asset	MT	WP3	"Derived from P4 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 1 (Security Investment), Use-case 2 (Audit) and Use-case 5 (Quick scan), specifically U1.11, U1.17, U2.9 and U5.11."	
R78	The TRESPASS tools should be able to calculate the effect of differences in the model on the risk values	MT	WP3	"Derived from P5 (TRESPASS tools should be able to rank attack scenarios in a risk-based prioritisation). Supports Use-case 1 (Security Investment) and Use-case 4 (Product-service system), specifically U1.2 and U4.4"	
R79	The TRESPASS tools can calculate the effectiveness of a countermeasure in a navigator map from the map, a set of attacker profiles, and a cost function of the countermeasure.	MT	WP3	"Derived from P6 (TRESPASS tools should be able to calculate the cost-effectiveness of a proposed countermeasure). Supports Use-case 1 (Security Investment), Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U1.2 U2.11, U4.2 and U5.13"	

Table A.3.: Requirements inside WP3

#	requirement	source	target	goals	priority
R38	Attack tree generation from socio-technical model	WP3	WP3	This is an indispensible part of the intended tool chain. This is needed for task 3.4	

B. 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.

B.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.

B.2. Overview of TRESPASS Integration

The TRESPASS workflow involves several stages with various activities, some of which are optional. Figure B.2 shows the architecture diagram and Figure B.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.

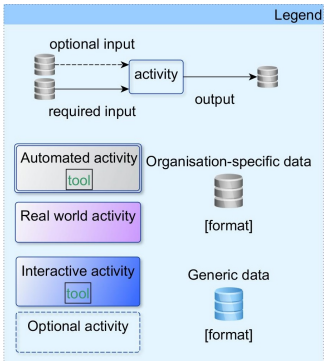

Figure B.1.: Legend for the Integration diagram in Figure B.2.

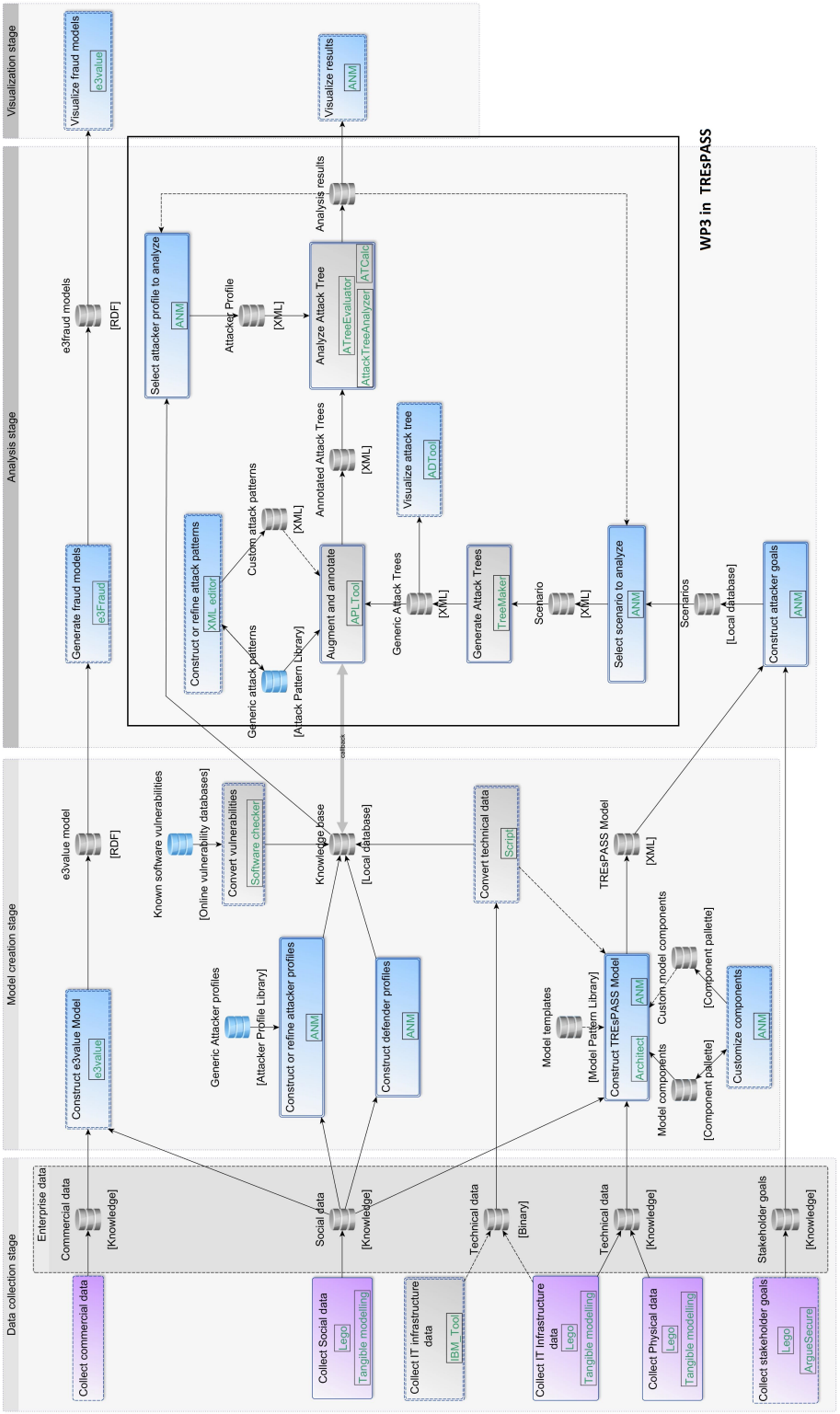

Figure B.2.: Integration diagram for the TRESPASS project.