predict
prioritise
prevent

TRE$_S$PASS

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

Deliverable D5.4.2

The integrated TRE$_S$PASS process

Project:	TRE$_S$PASS
Project Number:	ICT-318003
Deliverable:	D5.4.2
Title:	The integrated TRE$_S$PASS process
Version:	1.0
Confidentiality:	Public
Editor:	Margaret Ford, Jan Willemson
Cont. Authors:	A. Lenin, O. Gadyatskaya, D. Ionita, W. Pieters, A. Tanner, S. Saraiva, C. Muller, J. Willemson, M. Ford, S. Muller
Date:	2016-10-31

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	Dan Ionita	2, 7, 10
CYB	Aleksandr Lenin	5, 8
	Jan Willemson	1, 13
IBM	Axel Tanner	7
TUD	Wolter Pieters	12
UL	Olga Gadyatskaya	2, 4, 9
CHYP	Margaret Ford	All
GMVP	Sergio Saraiva	3
ITR	Cédric Muller, Steve Muller, Carlo Harpes	1, 5, 11

Quality assurance		
Role	Name	Date
Editor	Margaret Ford	2016-10-31
Editor	Jan Willemson	2016-10-31
Reviewer	René Rydhof Hansen	2016-10-15
WP leader	Jan Willemson	2016-10-31
Coordinator	Pieter Hartel	2016-10-31

Circulation	
Recipient	Date of submission
Project Partners	2016-09-30
European Commission	2016-10-31

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 (TRE$_S$PASS). 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

Management Summary

The TRE$_S$PASS project is concerned with socio-technical security risk analysis, predicting, prioritising and preventing security risks. To support this aim, a wide range of techniques and processes has been developed across the technical work packages (WP1 – WP4) within the project. The aim of WP5, and particularly this task T5.4 within WP5, is to incorporate these different methods into an integrated suite of processes for practitioners.

This deliverable describes the final development stage of the TRE$_S$PASS process as released by the end of M48 of the project.

Key takeaways

- The project has developed a range of complementary risk management processes and tools across its core areas of data collection, modelling, analysis and visualisation;
- These processes and tools have been developed and extended in collaboration with the case study owners e.g. the e3fraud modelling approach has been developed to suit the commercial requirements of the Telco case study and the TRE$_S$PASS model has incorporated specific features to support the modelling of virtualised environments for the cloud case study;
- Ongoing validation has been a core feature of TRE$_S$PASS process and tool development through practitioner workshops, surveys regarding the threat landscape and ongoing case study feedback;
- TRE$_S$PASS processes have been evaluated for their potential contribution in the context of the Estonian ISKE framework, CORAS, FAIR, and TRICK Service;
- The TRE$_S$PASS service model has been developed based on the Attack Navigator concept to allow for seamless incorporation of client organisations.

1. Introduction

In this document, we present the elements of the TRESPASS process in the ways in which they have been successfully combined to achieve different outcomes within the project. We present the work of the technical work packages: data collection (WP2), modelling (WP1), analysis (WP3) and visualisation (WP4). We then proceed to describe the application of the TRESPASS process in relation to our case studies: IPTV, Telco, Cloud, ATM, and personal data protection. We investigate the potential role of TRESPASS in the wider risk landscape, through an analysis of the ways in which it might be compatible with or complementary to a major existing framework. For this analysis we have selected the ISKE framework, which derives from the German BSI framework and is widely implemented in Estonia, and CORAS, FAIR, and TRICK Service as three of the outstanding established risk assessment frameworks. As a new component, we will also present the TRESPASS service model.

This is the second deliverable for this task, which relies on all deliverables to date from work packages 1 – 5. An important related document is Deliverable 6.2.2 (The TRESPASS Project, D6.2.2, 2015), which presents a project-wide view of requirements. Deliverable 5.1.2 (The TRESPASS Project, D5.1.2, 2015) and Deliverable 6.1.2 (The TRESPASS Project, D6.1.2, 2015) are the requirements deliverables for WP5 and WP6 respectively. These provide useful background reading, highlighting the role of WP6 purely in relation to digital tools, while WP5 encompasses the full range of processes within the project.

1.1. Motivation and challenges

The aim of this document is to present the processes developed in the TRESPASS project as an integrated whole. The various work packages have produced a diverse range of processes and tools for practical application. This document highlights the ways in which these processes and tools can be used in the context of providing decision support for security practitioners. The aim is to provide the practitioner with the appropriate tools to evaluate the different possible courses of action within their own given resource constraints.

Within this document, several different approaches are explored and evaluated with reference to a number of different use cases. This is in line with the project decision, underpinned by feedback from practitioners, the advisory board and reviewers over the course of the project, to adopt a reasonably loosely-coupled approach to both processes and tools. Appropriate tools may be selected from the project toolkit according to the immediate needs of the practitioner. This allows the project to propose a more flexible range

of options when applying processes in organisations of all sizes, including public sector organisations, large corporates and SMEs.

1.2. Goals

In describing the TRESPASS tools and processes within this document, the goal is to identify the ways in which a practitioner may derive value from their application. This involves the exploration of different scenarios and the appropriate application of processes and tools to those scenarios. It is not intended that there should be a single end-to-end TRESPASS process, but rather that elements of the TRESPASS toolkit should be applied as appropriate to a given environment. This may involve the application of a single specific process, or a number of related processes together in order to achieve a more effective result.

1.3. Structure of the document

Within this document we present an outline of the types of tools and processes developed within TREsPASS and a description of the ways in which these have been used in the different case studies within the project. We then continue to consider how these might be applied in other contexts, including their relationship to major security risk management frameworks, such as ISKE in Estonia, CORAS, FAIR and TRICK sercice developed by partner itrust. We finish the document by presenting the details of the TRESPASS service model.

1.4. Foreground and background

As an evaluation of the application of the tools and processes developed within TRESPASS to different scenarios, the majority of this deliverable may be considered as foreground. Where appropriate, reference is made to other TRESPASS deliverables which describe the tools and processes concerned in more detail. These other deliverables may also contain greater detail about the relative balance between foreground and background in each case.

2. TRESPASS Process Overview

In this chapter, the different processes which have been developed within the TRESPASS project and their uses in the risk assessment process are described. These processes have been developed in the technical work packages and subsequently validated by means of a number of practitioner panels, held in different countries and with different practitioner communities. The practitioner panels were run using provocations and stimulus material from the Attack Navigator Prototypes, as described in Deliverable D4.1.2 (The TRESPASS Project, D4.1.2, 2015). The work on attack trees, visualisation of cloud scenarios and the attack navigator map were also presented, along with recent tools and prototypes. The feedback from these panels has provided valuable information for process development across the project.

In addition to validation through the practitioner panels, different aspects of the TRESPASS process have been validated by their application to the project case studies. In the first year of the project, the main focus was on the IPTV case study which looks at making card payments via the TV remote. The Telco case study, which focuses on fraud within an international telecoms environment, and the Cloud case study, which investigates the particular challenges associated with virtualised environments, have subsequently been increasingly prominent. Additionally, a newer case study on securing ATMs has been developed to explore physical, logical and geographical elements of security in greater depth.

There are a number of key stages involved in the TRESPASS risk assessment approach. A comprehensive outline of the associated workflow is depicted in Figure 2.1.

The four main stages in the TRESPASS process are data collection, modelling, analysis and visualisation. Data collection is vital to understanding the nature of a scenario and providing input to the subsequent tasks of modelling, analysis and visualisation. Within the project, the focus has been on the 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 perspectives on the nature of the organisation being investigated.

In the area of modelling, approaches have been adapted according to the needs of a particular environment. The 'Level Zero' modelling approach (i.e. an approach making no assumptions on the previous models or modelling attempts) has enabled an early mapping of the wider environment, in order to understand where further investigation may usefully be targeted.

The TRESPASS model provides a means of describing an environment and the possible attacks on it in greater detail. While this takes a largely architectural view of the organi-

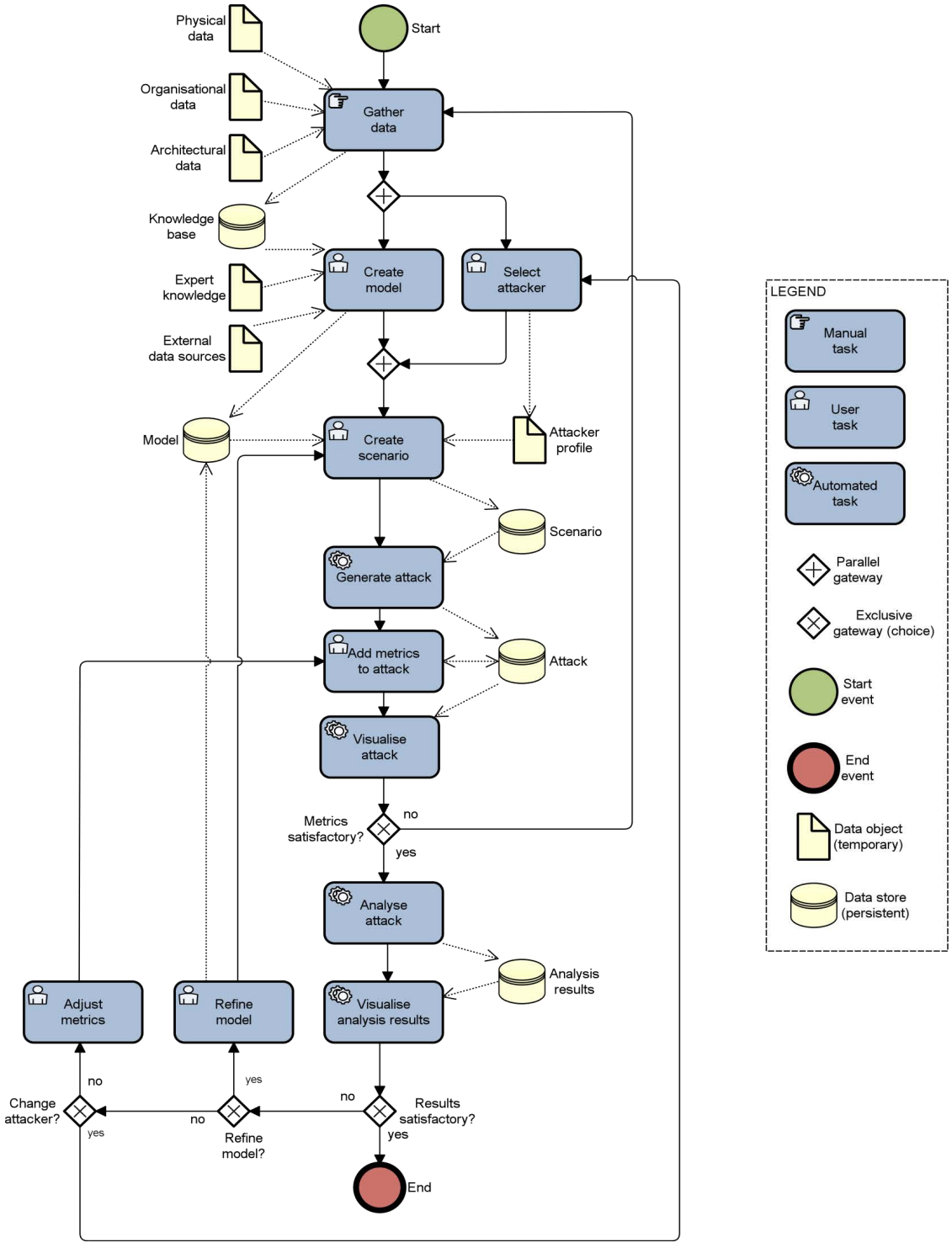


Figure 2.1.: TRESPASS workflow diagram

sation, the e3value method has been adopted to explore the contractual and commercial

relationships in more detail. The QGIS tool is being used in the ATM case study to model the geographical locations of Automated Teller Machines and evaluate to what extent specific features of their location are associated with reports of adverse events.

A range of analysis methods has been developed within the project, with the ability to provide answers to key questions such as the most cost-effective or quickest means of achieving a successful attack. Many of these are supported by tools such as the ATree Evaluator tool, which provides a means of evaluating trade-offs between different parameters as a Pareto frontier. These methods are each designed with different requirements in mind, although in some instances they may be used in combination.

TRE$_S$PASS has also developed innovative approaches to visualisation. The project has focused particularly on visualising elements of the analysis, as this is key to the overall project goal of providing 'decision support' to practitioners. However, visualisation has also proved valuable in many other contexts, including data gathering and model development.

Some of these processes are manual, while others are digital and may be automated to varying degrees. The tools relating to the digital processes are represented in the architecture diagram, described in Deliverable D6.2.2 (The TRE$_S$PASS Project, D6.2.2, 2015).

2.1. Data collection stage

The data collection process is described in more detail in Deliverables D2.2.2 (The TRE$_S$PASS Project, D2.2.2, 2015) and D2.3.2 (The TRE$_S$PASS Project, D2.3.2, 2015), which focus on technical and social data respectively. Subsequent deliverables will focus on the integration of digital, social, physical and commercial data, as well as handling data which is of inconsistent quality. One or more of the following kinds of data may be required as input to the subsequent modelling, analysis and visualisation processes.

2.1.1. Technical data

Data regarding the digital assets of an organisation may be collected manually or digitally, depending on the type of data and available sources. A description of the IT infrastructure of the organisation provides essential input for the construction of the TRE$_S$PASS model and/or subsequent analyses. In addition, policies regarding the handling of digital data and log files showing the history of transactions across a system provide useful input to the risk management process. The policy information is more likely to be created manually, whereas log files are almost invariably created as part of a digital process. Gathering data regarding digital assets can result in greater knowledge about the system concerned, as well as providing specific files containing relevant information such as diagrams, logs and reports. The Cloud case study has explored the gathering of digital data, both manually and automatically, in some depth. In particular, the specific requirements of a highly

virtualised environment have presented significant challenges. In order to accommodate these requirements, a number of extensions to the original TRESPASS model and associated processes have been introduced.

2.1.2. Social data

Many different techniques have been adopted in order to gather social data within the project. Some of these, such as soft systems methodology, as described in Deliverable 2.3.1 (The TRESPASS Project, D2.3.1, 2014), focus on social data at the organisational level. This provides a very valuable basis for building the model of an organisation. Some of the more statistical approaches associated with crime science focus more on social data at the individual level. The knowledge gained from this manual, real-world activity contributes to the creation of profiles for the actors referred to in the TRESPASS model and/or required by the subsequent analyses. This may include: attributes of targets (such as employees), stakeholders, clients and potential attackers. These different approaches to gathering social data have been explored in considerable detail in the context of the IPTV case study. The relationship between the case studies and the developed processes is explored at some length in subsequent chapters of this document.

2.1.3. Physical data

Physical data is normally collected manually, providing knowledge regarding the physical layout of an organisation. This data may subsequently be used to construct a TRESPASS model and/or as input to one of the analysis processes. This physical data includes items such as locations, buildings, rooms, doors, windows. Physical data forms an important aspect of the ATM case study, which is being explored in considerable detail in the final year of the project. Specifically, the location of an ATM, both within a building and in relation to the local area, may be used to determine the most likely types of attack and appropriate means of defence.

2.1.4. Commercial data

Commercial data is required for input to processes such as the *e3fraud* analyses. An e3fraud analysis can be conducted independently from all other TREsPASS analyses and is focused on identifying the possibilities for fraud that a particular set of commercial agreements is vulnerable to. Thus, the data that needs to be collected during this manual activity includes, but is not limited to: contractual agreements, service level agreements, pricing schemes, operational costs and market estimates. The gathering of commercial data has been particularly important for analysing business aspects of the Telco case study, which focuses on the opportunities for fraud within a complex global network of contractual agreements.

2.1.5. Stakeholder goal

The purpose of this manual activity is to identify assets and policies to be protected, which are considered critical to one or more stakeholders. This information is only required if custom attack scenarios are to be analysed.

2.2. Model creation stage

The different types of data described above may be imported to a knowledge base, for subsequent use in the model creation and analysis processes. It should be noted that the model is intended to be relatively lightweight, compared with the analysis techniques. The knowledge base provides a means for the analysis techniques to request data, based on the structure of the model, in order to populate their respective calculations. Where the analysis results need further refinement, it may be necessary to return to the data gathering phase to elicit further data in an iterative process.

The model creation stage handles the creation of the TRE$_S$PASS model and associated actor profiles, including both attackers and defenders. The e3value model creation process is complementary to the main TRE$_S$PASS model, for cases requiring a more specifically commercial focus. The QGIS mapping process provides a means of incorporating geographical data into the standard TRE$_S$PASS risk assessment approach.

2.2.1. TRE$_S$PASS model creation

This is a key activity in the TRE$_S$PASS process. Its target is to output a system model that can be further extended and analysed. Creation of the TRE$_S$PASS model is supported by the use of the respective tools, e.g. Architect for ArchiMate modelling and conversion to the TRE$_S$PASS model, or the special component dedicated for this purpose in the Attack Navigator Map toolset.

Optionally, customisation of components may take place before or during the TRE$_S$PASS model creation and allows the user to create custom model components, which can be saved in the knowledge base for later use.

2.2.2. Profile creation

Profiles are required within the analysis process, to identify key attributes of the actors under investigation. Attacker profiles may contain key information such as skill level or financial resources. The defender or target profile may identify an individual's role or personal characteristics.

Attacker profile creation The TRE$_S$PASS model analysis is typically relative to a specific attacker profile, since omnipotent attackers are not only unrealistic but also give unreasonably conservative analysis results. We envisage that there will be a finite set of ready-made attacker profiles stored in the attacker profile library and the user will be able to select and instantiate a relevant set of these via the Attack Navigator Map toolset.

Defender/target profile creation In order to decide the type of results which different elementary attacks will have on any given target, the attack generation process needs to have access to the target parameters. These constitute the target profile, which is developed based on the social data gathered in the data collection phase.

2.2.3. e3value model creation

This interactive activity involves using the *e3value toolkit* (Gordijn, Akkermans, Koks, & Schildwacht, 2004) to create *e3value* models (Gordijn & Akkermans, 2001). These models structure the commercial data gathered in the data collection stage in a formal way, so that it can be analysed automatically.

The e3value model creation process is in detail described in Deliverables D7.3.1 (The TRE$_S$PASS Project, D7.3.1, 2014) and D7.3.2 (The TRE$_S$PASS Project, D7.3.2, 2016).

2.3. Analysis stage

2.3.1. TRE$_S$PASS analysis

A number of different steps are involved in the TRE$_S$PASS approach to analysis. This section specifically describes the individual steps in the overall process supported by the TRE$_S$PASS toolkit, described in Deliverable D6.4.2 (The TRE$_S$PASS Project, D6.4.2, 2015).

2.3.1.1. Attacker profile selection

To start the analysis (or more specifically, the attack generation), the TRE$_S$PASS toolset needs to have a defined attacker profile in place. In this step the user can select the attacker profile for input to the analysis. If multiple profiles need to be analysed, the analysis must be re-run for each one. It is likely that once a particular analysis has been run, a different profile may be selected to compare its effect on the analysis results.

2.3.1.2. Attacker goals creation

In addition to the attacker profile, the TRESPASS toolset needs to have defined attacker goals in place. These can be developed manually from the stakeholder goals or derived automatically from the selected attacker profiles.

2.3.1.3. Scenario selection

Each TRESPASS analysis can be run against a single pairing of attacker and attacker goal. We call such a pair a *scenario*. After selecting a particular attacker profile, and creating or generating a set of possible goals for that attacker, it is now necessary to select a specific goal to define a particular scenario.

2.3.1.4. Attack pattern creation and sharing

Automated attack generation can only reach a certain level of abstraction, which may not be detailed enough for quantitative analysis. In order to expand the generated attack tree further, additional attack patterns may be needed. It is intended that these patterns should be stored in a library for ease of access. This library can be created manually and stored locally in the knowledge base. Where desired, the library may also be shared with other risk analysts who in turn can use these patterns as a ready-made library for their own analyses.

2.3.1.5. Attack generation

This is the core analysis step that generates possible attacks from the system model. For instance, an attack tree may be generated automatically by the Treemaker tool from the TRESPASS model. This process is described in more detail in Deliverable D3.4.1 (The TRESPASS Project, D3.4.1, 2014).

2.3.1.6. Attack tree annotation, augmentation and pruning

There are special components which can augment the generated attack tree with the pattern trees stored in the Attack Pattern Library via a callback to the TRESPASS Knowledge Base (see Deliverable 2.4.1 (The TRESPASS Project, D2.4.1, 2016)). As a result, we will have a tree with decorated leaf nodes suitable for quantitative analysis. The leaf parameter values are also obtained via a callback to the data gathered previously and stored in the Knowledge Base.

2.3.1.7. Attack tree analysis

Several useful quantitative questions can be asked based on the attack tree, e.g. what is the attack with the best expected utility or what is the success probability of the attack. In order to answer these questions, a number of methods and tools have been created (The TRESPASS Project, D3.3.2, 2015). These include a tool which returns Pareto-optimal values with maximum probability and minimum cost for an attack and a specialised process which uses time dynamic analysis to estimate the amount of time that an attacker will take to penetrate a given system.

2.3.1.8. Attack tree sharing

Attack trees are a succinct knowledge representation format describing attacks on systems. Therefore, it can be very beneficial to share them. An attack tree sharing process is implemented via the Attack Tree Library. The tree sharing process is not contextual. Any attack tree can be shared by the user without considering its sensitivity, and any attack tree can be retrieved from the library. A methodology for contextual attack tree sharing is presented in Deliverable D5.3.2 (The TRESPASS Project, D5.3.2, 2015). The interested reader can refer to this deliverable for more details about attack tree sharing, or model sharing in TRESPASS in general.

The Attack Tree Library is prototyped as a git repository on the gitlab server[1]. At the moment this repository has the following features:

- **Open submission of new trees and access to available trees:** Any user with an account on gitlab can access the available trees, represented as XML files, and submit new models to the repository.
- **Standard format of the trees:** The ADTool XML schema is used as a standard format for trees. It supports attack trees and attack-defence trees.
- **Tree visualisation capabilities:** Visualisation is available via the ADTool.

2.3.2. *e3value* analysis

In order to generate possible fraud scenarios, the user needs to indicate which actor in the e3value model provided is the Target of Assessment, as well as an interval of expected occurrence rates of the commercial transactions specified by the e3value model.

The e3fraud tool generates all possible ways in which the e3value model given as input can go wrong: payments might not take place, hidden payments might occur and actors can collude. It also ranks these models based on loss for a particular actor (the actor whose perspective we take) and on the delta in profit of the other actors in the model when compared to the e3value model given as input. The higher the loss for the main

[1] https://gitlab.com/twice3/Attack-Tree-Library.git

actor, and the higher the gain difference for any other actor, the higher the rank of the model.

2.4. Visualisation stage

Visualisations may be used at a number of different points within the TRESPASS process to provide practitioners with feedback regarding the results of their activities. In this respect, the model creation, attack generation and analysis results are key points at which visualisations could be used to support further decision-making. The TRESPASS approach to visualisation is described in detail in Deliverable D4.3.1 (The TRESPASS Project, D4.3.1, 2014). It should be noted that visualisation also forms an important aspect of the user interface, as described in Deliverable D4.2.1 (The TRESPASS Project, D4.2.1, 2014).

2.4.1. Attack tree visualisation

In Deliverable D4.3.1 (The TRESPASS Project, D4.3.1, 2014), possible approaches to visualisation are described in considerable detail. Starting from the inherent features of a tree structure, it continues by exploring the visualisation capabilities of the ADTool, developed by project partners UL. The deliverable then goes on to investigate the relationship between visualisation and modelling in a wider context, and the influence which form can have on perception of data. Deliverable D4.2.1 (The TRESPASS Project, D4.2.1, 2014) looks in some detail at possible approaches to visualising attack trees, showing different ways to represent relationships and layering, according to the environment being represented. Representations of key aspects such as difficulty, time and probability are shown in different ways, including increasing line width, colour and intensity. Additionally, the visualisations present scenarios as attack trees in many different forms and from many different perspectives, highlighting the features which are likely to be of most interest in a particular context.

2.4.2. Attack tree analysis visualisation

Given the diverse analysis tools available, it has been necessary to specify the desired visualisation outputs in considerable detail. The features of the different tools have been analysed, as described in Deliverable D4.1.2 (The TRESPASS Project, D4.1.2, 2015), in order to identify the most appropriate means of visualising both the features that are common across the tool set and the individual features in each which may be of particular value to a practitioner.

2.4.3. Fraud model visualisation

The e3fraud tool shows the generated attacks as a ranked list of textual descriptions of the form:

- Payment X does not take place.
- There is a hidden payment between User A and User B.
- User A and User C are colluding.

and can also display charts showing the profitability for each actor in these generated models.

2.5. Step-by-step guide

This section will present the step-by-step guide through the TRESPASS process as supported by the Attack Navigator Map (ANM).

1. First, the user directs the web browser to the ANM installation and presents the login credentials (see Figure 2.2).

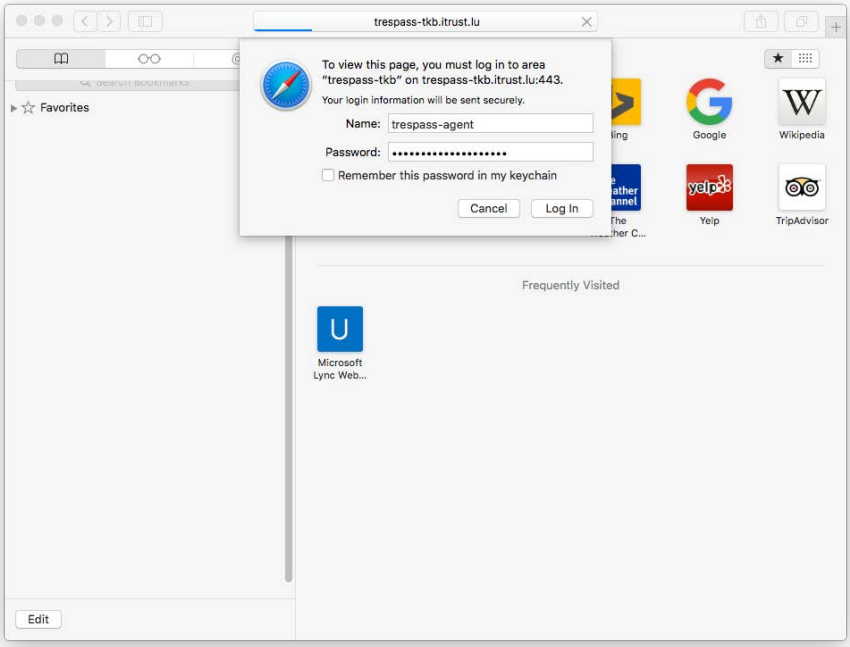


Figure 2.2.: ANM login screen

2. The user is then presented with the working canvas (see Figure 2.3).

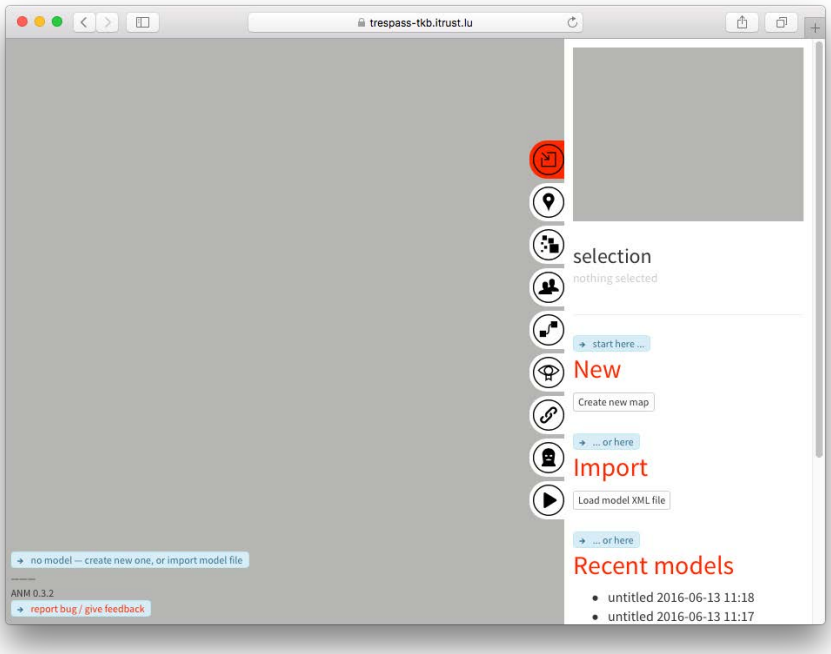


Figure 2.3.: ANM working canvas

3. The user creates a new model by pressing the "Create new map" button (see Figure 2.4).
4. The user can populate the model using predefined model patterns (see Figure 2.5).
5. The user will add the actors and assets, selecting their types and properties (see Figure 2.6).
6. The user will select the attacker goal together with its value, select the tool chain and run the analysis (see Figure 2.7).
7. The analysis runs following the selected toolchain (see Figure 2.8).
8. If no specific knowledge base component is installed, the first analysis is run using default augmentation and leaf node evaluation which is probably not what the user wants in the long run. TRE$_S$PASS toolset offers a very flexible approach for the user to define one's own attack tree decorations. For that, the knowledge base files may be modified directly. Going back to the canvas screen (see Figure 2.7), the user may select the "edit knowledgebase files" link from the lower left corner. The user is then taken to the file browser (see Figure 2.9).
9. The user opens the `apl_logic.py` file and modifies the default augmentator and annotator methods (see Figure 2.10).
10. The user runs the toolchain again.

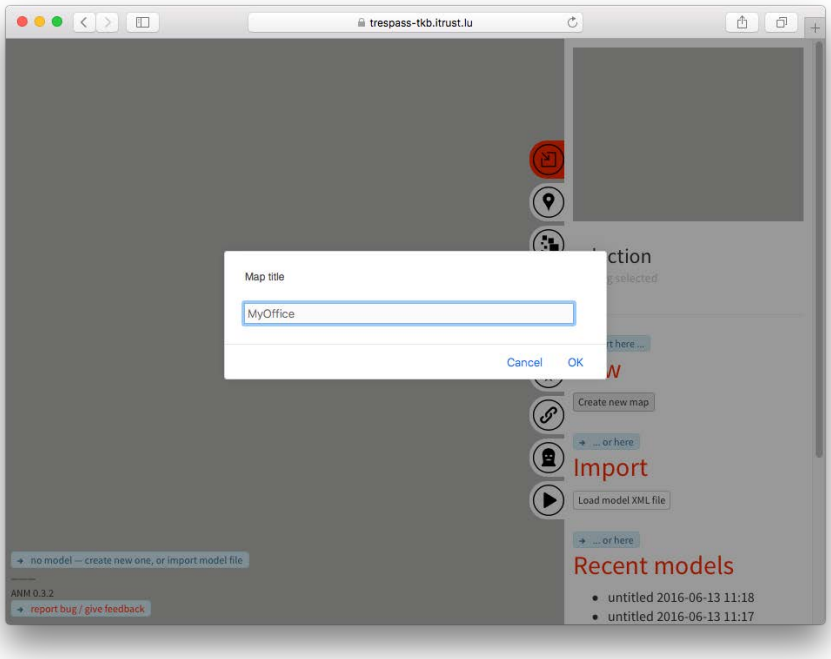


Figure 2.4.: ANM new model creation

2.6. Conclusions

In this chapter we have outlined the different elements which together constitute the TRE$_S$-PASS process: data collection, modelling, analysis and visualisation. In the following chapters we will describe these in greater detail, as well as looking at their application to the TRE$_S$PASS case studies and prominent industry frameworks: ISKE, CORAS and FAIR.

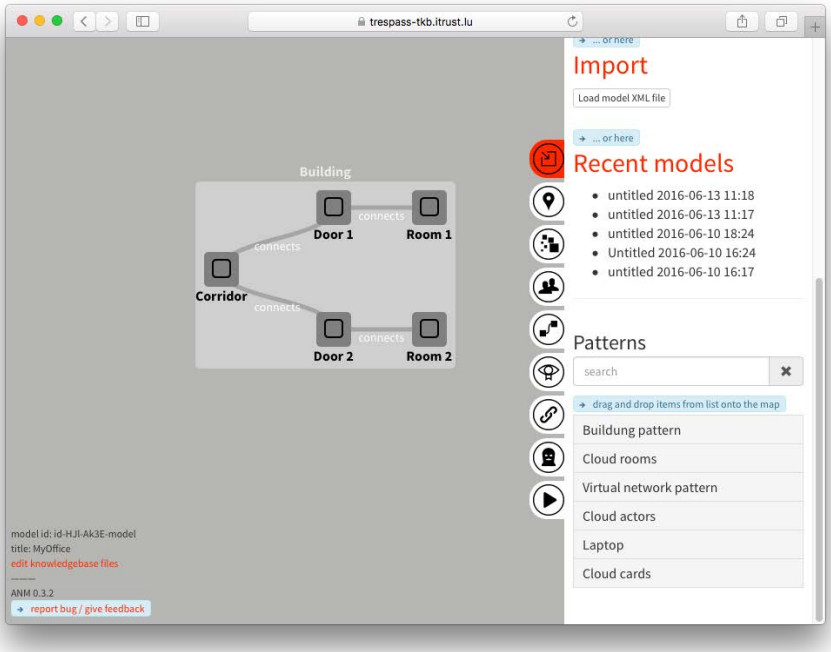


Figure 2.5.: Using model patterns

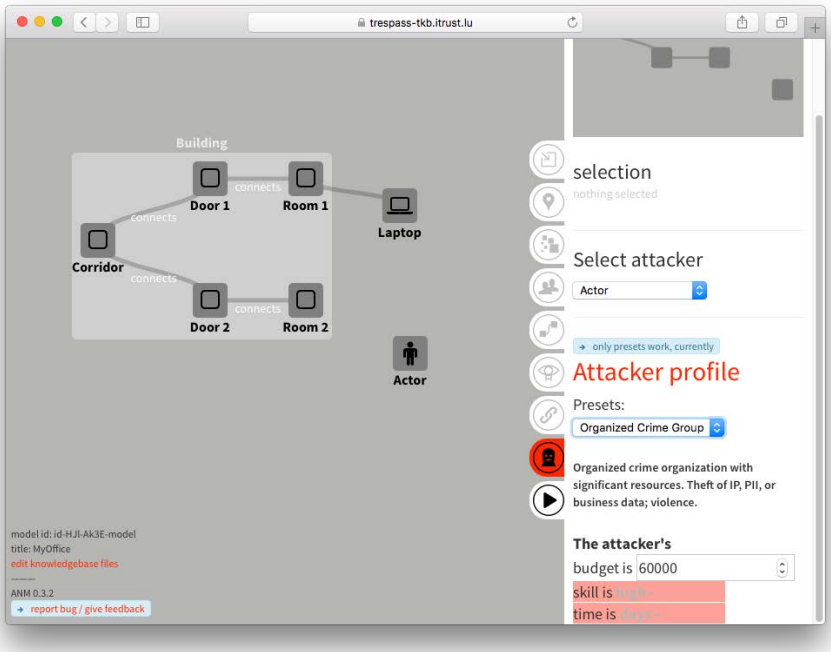


Figure 2.6.: Adding actors and assets to the model

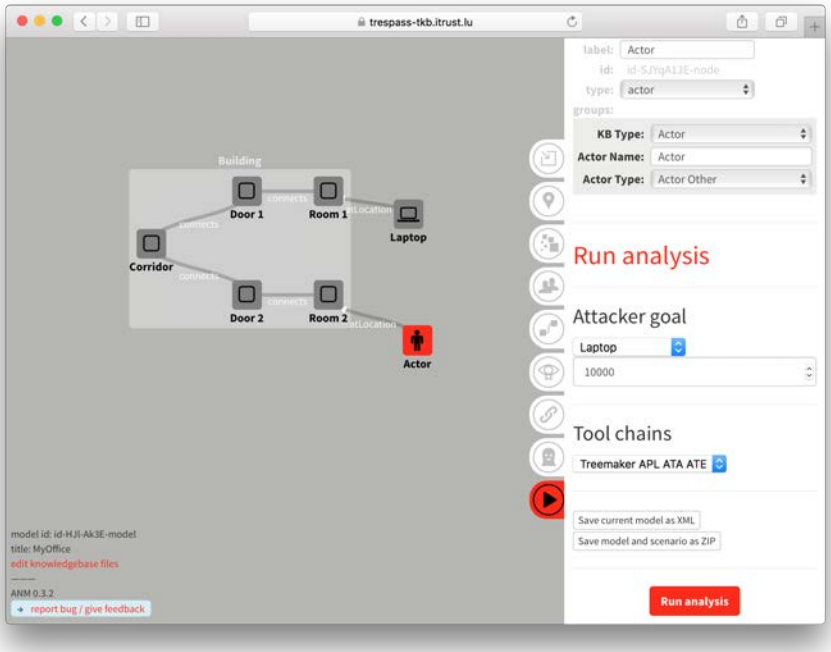


Figure 2.7.: Running the analysis

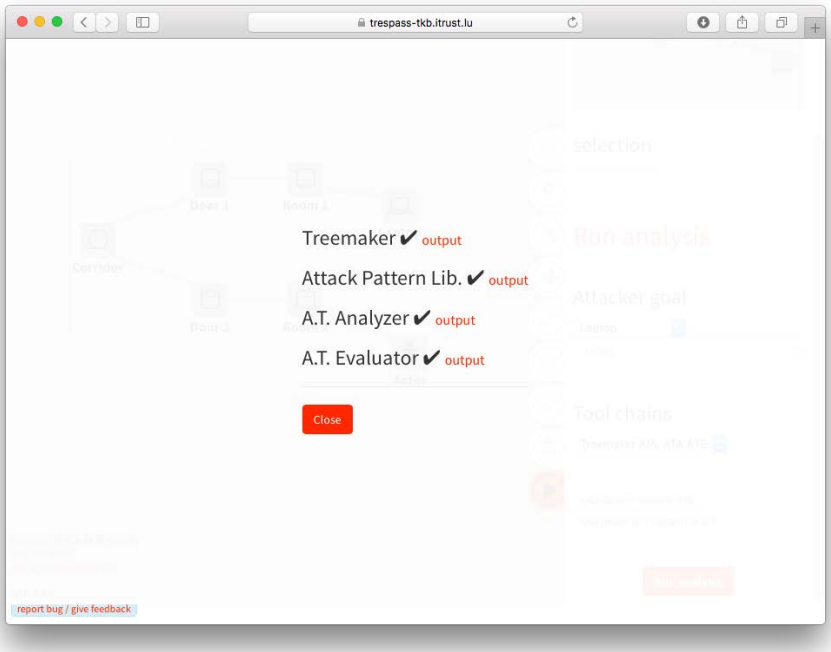


Figure 2.8.: The first run of the analysis toolchain

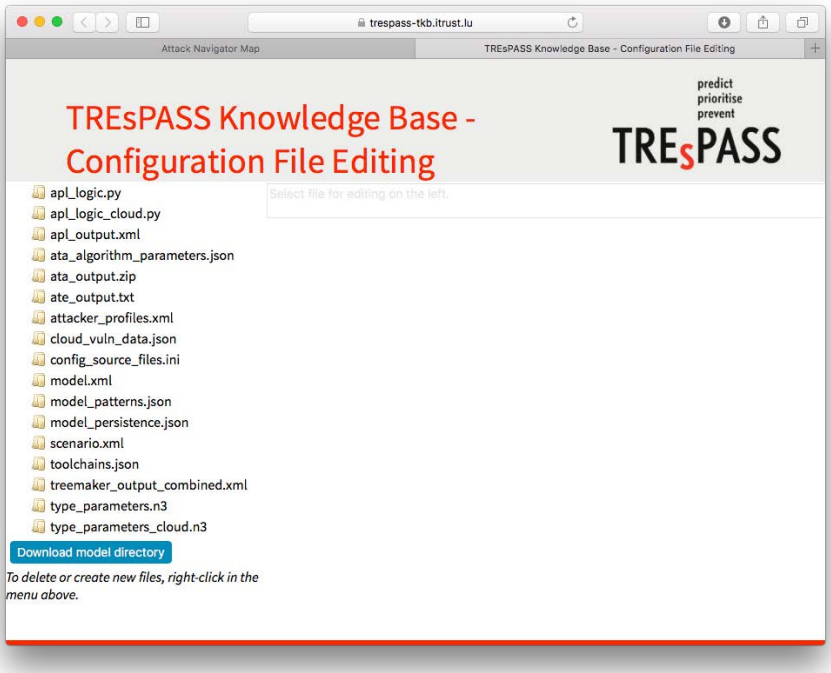


Figure 2.9.: Knowledge base file browser

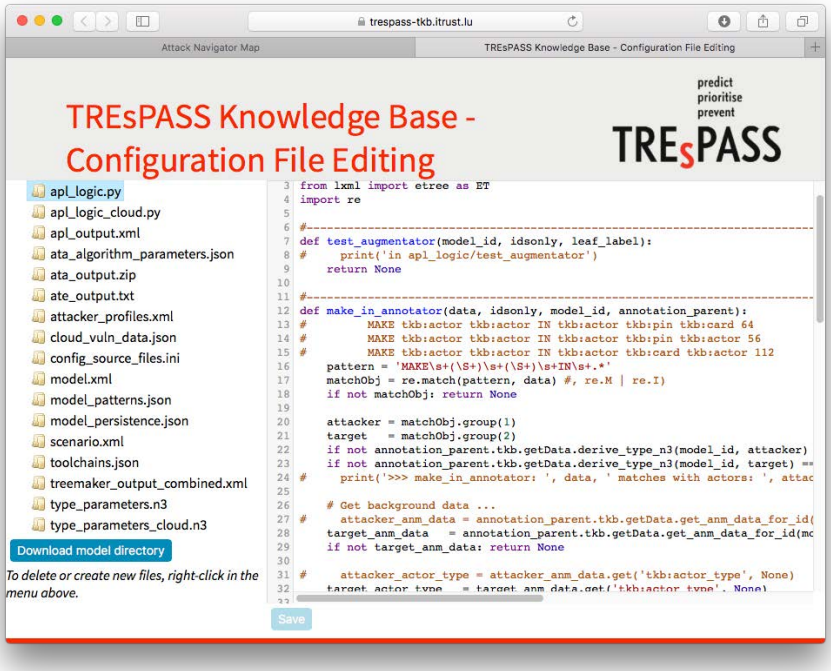


Figure 2.10.: Editing the knowledge base

3. Data Collection

In this chapter we look more closely at the data collection process within TRESPASS. The constituent processes are presented according to the type of data being collected. Further information on the handling of technical data is available in Deliverable D2.2.2 (The TRESPASS Project, D2.2.2, 2015), while the collection of social data (including commercial data and physical data) is described in Deliverable D2.3.2 (The TRESPASS Project, D2.3.2, 2015). The aim of the data collection process within TRESPASS is to provide appropriate input to the modelling, analysis and visualisation processes, in order to produce high quality results which will be useful to practitioners.

A number of possible data sources were identified in Deliverable D2.2.1 (The TRESPASS Project, D2.2.1, 2013), which emphasises the distinction between *infrastructure data*, *attacker data*, and *real-time data on actions*:

- Quantitative penetration testing;
- IDS/IPS systems;
- Security and event logs;
- Perimeter security elements (i.e. firewalls conducting deep packet inspection);
- Social science experiments in social engineering;
- Social science inductive research, which may establish patterns for data protection practice;
- Historical data from EU institutions (i.e. Europol);
- Information about known security incidents;
- CCTV surveillance systems.

3.1. Technical data

The TRESPASS approach to handling technical data is composed of the following steps:

- Discovery - the process of exploration of an environment in order to identify the range of available data. It enables the discovery of potential data sources. The data discovery and extraction processes must be designed to be sufficiently versatile to work with the data sources available in the environment where the TRESPASS tools are implemented.

- Extraction - the process of getting data suitable for supporting risk calculations from an organisation. Where appropriate for the data type, the data extraction must be as highly automated as possible.
- Analysis and integrity checking - the process of categorisation, transformation, verification and evaluation of the data. Consolidation also fits here.
- Storage - the process of defining the data format and storing the data properly.
- Output - How the data will be shared or exchanged with other parties.

State of the art knowledge is available regarding the extraction of data from digital sources, as applied to information security. For instance, intrusion detection and prevention systems (IDS, IPS) collect traffic information to identify any evidence of an attack. Security information and event management (SIEM) tools also provide a central repository for logs which may help to identify evidence of attacks. The TRE$_S$PASS approach builds on these processes and extends them where necessary to meet the particular needs of complex socio-technical risk environments.

As an example application of such an approach, a tool for data extraction from virtualised environments has been created in TRE$_S$PASS (see Deliverable D7.2.2 (The TRE$_S$PASS Project, D7.2.2, 2016)).

- The tool supports extraction of entities in the cloud environment like host systems, virtual machines, network and storage connections, together with user ids and their respective permissions on the infrastructure.
- The tool is geared towards extraction from a VMware environment with a centralised vSphere management.
- Additionally, a web-based visualisation tool provides an overview of the cloud environment with a focus on the access-control of the user ids of the system.

The tool is adapted specifically for centralised VMware infrastructures. Given a user id with suitable access rights, allowing read access for all items of interest in the vSphere environment, the script can extract data and will keep it as an internal RDF representation. From this RDF representation an export as GraphML can be generated that is in turn used to incorporate the information of the virtualised environment into the full TRE$_S$PASS model of a cloud environment.

3.2. Social data

Data extraction from social or physical environments typically involves audit reports, compliance checking, personnel records and interviews, based on previous security experiences, without objective grounding in data extraction processes. In this respect TRE$_S$PASS has progressed beyond the state of the art by developing tools which can be used to reflect upon and evaluate the quality of experiential rather than digital data. TRE$_S$PASS

focuses on the security of people by looking at the social practices that surround information exchange, by going back to the physical environments in which trust and resilience are built.

It is imperative to study the physical places and the social situations where security and security risks typically occur, as well as those where 'everyday' routines prevent such events from occurring. This is in order to understand not just how, why and as part of what social practices human error created a 'weak link', but where and how organisations have successfully avoided being made into the targets of attacks and where and how strong, resilient social networks may provide a natural protection of assets.

In Deliverable D2.3.2 (The TRE$_S$PASS Project, D2.3.2, 2015) the range of methods and data sets to be used to support the social dimensions of the Attack Navigator Map and undertake the risk calculations that use social data are described in considerable detail. The methods are presented in order of breadth of scale, starting with the broadest and progressing towards the finer detail:

- The tools and approaches start with aggregation techniques used in Geographical Information Systems (GiS) to provide a high-level overview of risk hot spots. The aggregation techniques that gather data through questionnaires and present the analysis using visual mapping techniques enable analysts to ground user behaviours and practices related to information sharing and protection of particular spaces. Traditionally patterns of practice are linked to physical spaces but in TRE$_S$PASS could also be developed to link to digital and organisational spaces.
- The next set of tools and approaches to be presented are those that form part of Stage Zero risk assessments. This involves participatory modelling techniques, designed to enable the different stakeholders to co-produce a model of the scenario. It also allows them to depict the different information-sharing and information-protection practices in operation within a particular scenario. Such modelling tools enable stakeholders to identify the goals and values of each community of practice, the potential conflicts between different information sharing and protection practices and the motivations behind information exchanges taking place within a scenario. The Stage Zero risk assessment approach can be usefully combined with the aggregation techniques to produce a more comprehensive map of information sharing and protection practices.
- Social engineering success stories is an innovative narrative technique that contributes attack technique data to the attack pattern libraries. Such a technique analyses attacker stories and produces patterns of attack steps and attack motivation weighted according to the frequency in the narrative corpus. This output can be used to overlay the information sharing and protection maps to better identify where there are gaps that might be exploited by attackers.
- Information sharing and protection maps can be enhanced in different scenarios through organisational records. The Call-Detail record technique illustrates how this can be done in the telecommunications scenario.

- Control strength is an important element of social data analysis. The practitioner survey of socio-technical cyber threats was used to illustrate how such surveys can be coded and analysed in varying ways to reveal different types of knowledge related to control strength and the threats to controls. These different coding and analysis approaches could be incorporated in the TRE$_S$PASS Attack Navigator Map. The Cues and Warnings and Security by Experiment techniques are particular approaches to testing the strength of controls and are valuable methods that can be used by security practitioners to refine a TRE$_S$PASS model."

The proposed TRE$_S$PASS approach to handling of socio-technical data will enable an analyst to gain a well-rounded overview of multiple and layered risk assessments of an ever-increasing number of potential scenarios, in support of key decisions. The variety of techniques enables the approach to be tailored to individual environments according to their context and requirements.

4. Modelling Processes within TRESPASS

Modelling is commonly used to support the evaluation of the technical infrastructure within an organisation, however the TRESPASS approach of incorporating social and physical aspects of security into the modelling process involves a significantly greater degree of complexity. The models need to have a strong, formal basis which is sufficiently substantial to support the needs of the analysis tools. At the same time, they must be sufficiently accessible and flexible to meet the varying needs of a busy practitioner community.

In this document we focus on the detailed modelling capabilities provided by the TRESPASS model and the complementary commercial capabilities of the e3value and e3fraud modelling approaches. The 'Level Zero' process described in the preceding chapter could be considered as a broad scale modelling process, used to map the wider risk landscape prior to focusing on a much more specific area to be modelled with the TRESPASS model.

4.1. The TRESPASS model

The TRESPASS socio-technical security model, as described in Deliverable D1.1.2 (The TRESPASS Project, D1.1.2, 2015), forms the basis for identifying possible attacks on an organisation. The TRESPASS model has a unique combination of features, which extend and complement existing models, making it an appealing option for adoption by researchers and practitioners. These features enable it to support comprehensive modelling of socio-technical aspects of an organisation, including physical buildings, technical architecture and social structures, both formal and informal. While it is common to model the digital architecture of an organisation, the addition of physical aspects is quite unusual and social aspects rarer still. The model employs entities, attributes and relations to provide a comprehensive view of the security environment within an organisation. A key feature is the containment of assets such as data and items, represented through location attributes. This containment can be arbitrarily deep. These features combine best practices from different models in a single, powerful method.

The TRESPASS model also supports processes and the associated actions and behaviour through the underlying process calculus, that has the capability to encode actor behaviour. The actors in the TRESPASS model are not explicitly treated differently according to whether they are insiders or outsiders. The insiders simply carry a greater level of knowledge about their environment.

The modular nature of the models is a significant strength, in that it allows for modular model development and maintenance, as well as compositional analysis of the models.

It also allows for different features to be added as required, according to the particular requirements of an individual case. Where models tend to be either textual or graphical, there is value in having elements of each. The TRESPASS model is based on XML, which enables it to be used as input to the analysis tools, as well as being visualised by the visualisation tools being developed within the project.

The handling of policies is an important aspect of the TRESPASS modelling process. In particular, the comparison of high level and low level policies may be instructive. Where there is a mismatch between policies, this may highlight an area in which an organisation is more vulnerable and so attacks are more likely.

The TRESPASS model supports quantitative annotations by using unique identifiers for all model elements. This enables the properties of those elements to be stored in the TRESPASS knowledge base for subsequent retrieval at the analysis and visualisation stages. These properties also provide a convenient means of handling vulnerabilities and countermeasures. A model API has been added, to enable the tools and analysis methods to obtain elements by type, identifier or connections.

The TRESPASS model is based on a new calculus, which represents actors as nodes that can move around and can contain other locations such as data; this unifies the way in which data is handled. Based on the model we have developed a new technique to systematically generate attack trees from system models; this is a novel formal underpinning of a process which is usually manual and informal. In addition, as a result of case study feedback, we have developed techniques for addressing elements of attack trees that are easy to recognise in attacks, but hard to generate from models. The model is able to express data derived from the case studies and can generate attack trees based on invalidating goal policies.

4.1.1. The TRESPASS modelling language

The model description language is based on XML, in order to ensure compatibility with other processes being developed throughout the project. It has a number of different elements:

- Locations: each location has an identifier and a domain. At present physical and network domains are supported. The physical domain is applicable to people, who may move around in their built environment. The network domain is applicable to digital programs/processes.
- Edges: edges describe the existing connections between locations. There is a source and target location associated with each edge.
- Assets: assets describe physical items or knowledge e.g. a password that an actor may possess. Assets are usually modelled as being in a particular location.
- Actors: actors may have assigned roles within the model. While each role is unique, several actors may fulfil that role.

- Predicates: predicates are used to describe relationships between actors. An example might be 'trusts' or 'is related to'. These offer flexibility in that they are not predefined.
- Policies: policies describe the conditions that an actor needs to fulfil to achieve a desired outcome. For example, a visitor may be admitted to a museum if they are carrying a valid ticket.
- Processes: processes run at a particular location. They receive particular inputs and, so long as the appropriate policy requirements are met, produce corresponding outputs.

4.2. Archimate

Within some of the TRE$_S$PASS case studies, the ArchiMate enterprise architecture modelling standard has been used as a front-end to construct socio-technical security models. A white paper published by The Open Group describes how the ArchiMate language can be extended for modelling risk and security aspects. This includes both concepts to support risk analysis and concepts to support security deployment, i.e. for specifying control objectives and (requirements for) control measures. The options described are, in decreasing order of preference:

- The use of ArchiMate concepts unmodified, as specified in the standard;
- The use of the extension mechanisms as specified in the standard to define additional attributes or specialisations of existing ArchiMate concepts;
- The use of additional concepts that do not yet exist in the ArchiMate standard and cannot be directly linked to existing concepts.

The white paper also proposes a "risk and security overlay" for the ArchiMate language, defining specialisations of both ArchiMate core elements and elements from the Motivation extension (the latter mainly for risk analysis), using the first two options listed above. Given the popularity of Archimate in the wider technology community, the ability to use it to interface with the TRE$_S$PASS model and tools is a valuable addition to the modelling process.

4.3. The Attack Navigator Map

The Attack Navigator Map (ANM) is a novel tool developed by the TRE$_S$PASS consortium. ANM consists of several components, but the first one of them displayed to the user in the toolchain is the modelling interface. Using this interface, the user may select from the palette of predetermined model components that map one-to-one with the TRE$_S$PASS model components. This makes creating a TRE$_S$PASS model very easy and natural, since the ANM will help to ensure that the resulting model file conforms to the TRE$_S$PASS

modelling language standard. At the same time, the full expressive power of the language is also made available to the user.

The ANM also provides a convenient way of creating higher-level models easier by offering the user a choice of ready-made model templates, grouping functionality and display enhancements. We refer to Deliverable 6.3.1 for a full description of the ANM as a modelling tool (The TRESPASS Project, D6.3.1, 2015).

4.4. e3value

The e3value approach has been adopted to handle the complex product packages promoted by telecommunication service providers, which rely on a wide variety of fees and agreements between different organisationas across the globe. While this environment could potentially be modelled using the standard TRESPASS modelling approach, the types of attacks which may be encountered in this environment are rarely related to the infrastructure of the telecommunications providers. The attacks encountered in this environment tend to stem from unexpected opportunities for attackers to combine offerings of different providers in different locations to achieve a profit. This activity is essentially parasitic, skimming revenue from one or other of the providers, without providing any tangible benefit. However, while one provider may be left out of pocket, another provider may have less reason to intervene if the part of the attack which involves them does not have a negative impact on their revenues.

An investigation of several known telecommunication fraud scenarios revealed that, unlike more technical attacks, they are best described in terms of value exchanges (such as money or services) amongst profit/loss responsible actors (The TRESPASS Project, D7.3.1, 2014). Since the fraud scenarios concerned were so substantially different from the types of attack initially envisaged in developing the original TRESPASS model, it was decided to adopt the e3value modelling approach, which is specifically intended to handle commercial aspects such as contracts and value transfer.

The original e3value toolkit had limitations for our risk assessment purposes, in that it only supported static analysis of an individual case. In order to compare two different cases, the ideal case and the non-ideal case, it was necessary to extend these capabilities. In addition, the degree of damage to the defender and the degree of profit for the attacker in Telco scenarios tends to depend on the scale of usage. If each call provides only a small profit, it will be necessary for the attacker to make a large number of such calls to achieve their aims. Equally, the loss to the provider will depend on the scale of the abuse. In order to handle these requirements, the e3fraud tool was designed to generate dual charts showing the dependency of the profitability with regard to usage, as well as the discrepancy between the ideal and sub-ideal models. The user may select the variables to be represented and the range of values to be included. The tool also supports the ranking and grouping of sub-ideal models based on several different criteria.

5. Analysis Processes within TRESPASS

In this chapter we list the available attack model analysis processes. More details about the analysis approaches developed by TRESPASS are available in Deliverable D3.2.1 (The TRESPASS Project, D3.2.1, 2015) and Deliverable D3.3.2 (The TRESPASS Project, D3.3.2, 2015).

5.1. Attack tree analysis processes

In this section we summarize the available analysis processes based on attack trees. Notice that attack trees, being the basic attack model in the project, appear in the general TRESPASS process as outputs of the Treemaker tool that generates them from the socio-technical model. The attack tree generation process is described in more detail in Deliverable D3.4.1 (The TRESPASS Project, D3.4.1, 2014). All approaches described in this section are compatible with the Treemaker attack tree format (TRESPASS .xml).

Analysis with the Failure-Free Model Tool This analysis method addresses the question whether an organisation is sufficiently secured against targeted profit-oriented attacks. It assumes so-called fully-adaptive adversaries that can re-run failed attacks an arbitrary number of times. The user needs to provide an attack tree and an attacker profile (containing such parameters as skill or budget) as input. The user may also specify control parameters of the genetic and adaptive genetic algorithm (the underlying computation engine).

Parallel model analysis with ApproxTree+ This analysis method is very similar to the previous one, but it assumes attackers that are not allowed to repeat failed actions again. As in the case of the failure-free model analysis, the user needs to provide an attack tree and an attacker profile as input, as well as optionally specifying control parameters for the genetic algorithm used as the underlying computation engine.

Analysis of attack trees with one parameter with ATree Evaluator This analysis method also focuses on the essential questions related to feasibility, probability and cost of successful attacks. The user needs to provide input in the format of an attack tree and the attribute values for basic actions, such as probability of success or cost of the action.

Analysis of attack trees with multiple values with ATree Evaluator ATree Evaluator also allows for more sophisticated analysis with incompatible objectives, such as success probability and cost (maximising probability and minimising cost), by means of multi-parameter optimisation. It reports Pareto-optimal values for the chosen objectives. The user needs to provide an attack tree as an input and two specific attribute values on basic actions: probability of success and cost of doing an action. In both cases the attack tree needs to be a binary tree in the TRE$_S$PASS .xml format.

5.2. Attack-defence analysis processes

TRE$_S$PASS also works on analysis techniques that are able to take into account elements of defender profiles and countermeasures. Below we summarize the analysis processes that include countermeasures and are able to guide the user towards defensive mechanism selection. Techniques presented in this section do not benefit from the possibility to use models automatically generated from the socio-technical system model. However, the project has investigated ways to implement an automated generation process for attack-defence models (e.g., attack-defence trees).

System and attack modelling using timed automata and model checking with UPPAAL This analysis method focuses on deciding whether there is a feasible attack given the time constraints. The user needs to provide a timed automata and (desirably) the timing constraints.

System and attack modelling using stochastic timed automata and statistical model checking with UPPAAL This method, based on stochastic analysis and simulation, evaluates different values for attack scenarios, such as probability estimates of attacks, hypothesis testing (e.g., statistical verification of countermeasures), and probability comparisons (e.g., determining the relative efficacy of one countermeasure against another). The user needs to provide probability distributions for the events of interest.

Single-parameter bottom-up computation of relevant attributes on attack trees and attack defence trees with the ADTool This analysis method is implemented in the ADTool, which contains 13 pre-defined attribute domains covering the most common questions regarding attack and defence scenarios, such as difficulty, minimal cost, maximal skill level, minimal time, and probability of success. The user is also able to define new attribute domains in the ADTool, and to perform single-parameter bottom-up computations on attack trees and attack-defence trees using these new attributes. As an input the user needs to provide an attack tree or an attack-defence tree (only trees designed in the ADTool or expressed in the ADTool XML schema are supported). The tree should have annotated leaf nodes with the desired attribute values.

Analysis of attack-defence trees with one or multiple parameters with ATree Evaluator ATree Evaluator is also able to handle computations on attack-defence trees with incompatible objectives. The user needs to provide an attack-defence tree and the attribute values for basic actions as an input.

Analysis of attack-defence trees with the ADTop ADTop (Attack-Defence Tree optimiser) is a tool for risk analysis which takes an attack tree, for instance designed with the ADTool, and an extract of risk analysis, for instance designed with the TRICK Service tool, as input information. As output, it provides sets of possible security controls in the form of optimal attack-defence trees thanks to libraries of countermeasures and an association matrix which makes the link between attacks and suitable security controls. It uses the Return On Security Investment (ROSI) concept to evaluate the usefulness of the generated attack-defence trees, and finds the most useful one(s). The node attributes used for this computation are the success probability of attacks, the effectiveness of defences and their implementation costs.

Summary The analysis step of the TRE$_S$PASS process incorporates a plethora of tools that can answer important questions about possible attacks on the system. To use the analysis processes, the user needs to provide an attack tree. In TRE$_S$PASS attack trees come after the modeling step, as they are automatically generated by the Treemaker tool from socio-technical system models annotated with data. After the analysis, the next step in the TRE$_S$PASS process is visualisation of the analysis results.

6. Visualisation Processes within TRESPASS

Visualisation is a very important aspect of the TRESPASS process, providing a means to highlight particular aspects of a risk scenario. The TRESPASS visualisation capabilities can also be integrated with other methods and tools within the project to support a greater degree of understanding of the risk environment and therefore a higher quality of decision-making. Each of the main steps within the process can be enhanced by visualisation:

- Data collection - visualisation can provide an insight into the nature and quality of the available data. This may be a digital visualisation of patterns within data held digitally, such as passwords or access control policies. It might equally be a paper-based or LEGO visualisation of an organisation under investigation.
- Modelling - high quality visualisations of the inputs to the modelling process can improve the practitioner's understanding of the material to be worked with. Given that the TRESPASS model has the potential to be presented in graphical form, visualising this in different ways can contribute to the handling of the model and its subsequent outputs to the analysis tools.
- Analysis - a considerable amount of work has been done within the project to develop a range of visualisations around the analysis tools. These started with approaches to visualising attack trees and have since been extended to take into account the nature of the individual tools. The subtrees could be shown on their own (potentially overlaid, producing a sort of heat map), or highlighted in the original attack tree. Attack traces annotated with values could also be visualised in radar / spider web graphs. Paper and digital prototypes will be used to evaluate which visualisations are most meaningful and accessible for particular user communities, as described in Deliverable D4.1.2 (The TRESPASS Project, D4.1.2, 2015).
- Visualisation can be used as a substantial decision support tool in its own right. Being able to compare different perspectives on a single environment provides much greater depth of understanding.

Three different aspects of visualisation are being explored concurrently within the TRESPASS project:

- Visualisation principles;
- Risk visualisations;
- End user preferences.

6.1. TRESPASS visualisation principles

The visualisation principles are applied throughout every aspect of the visualisations being developed in TRESPASS and have been developed through the use of digital and paper prototypes. The risk visualisations use the visualisation principles and apply them to the tools being developed within the project, especially the analysis tools being developed in WP3. The end user requirements are being evaluated through a series of workshops with different communities of practitioners, using digital and paper prototypes. Their feedback is used to shape the further development of the tools and interfaces being produced within the project.

The TRESPASS approach to process and tool development has been validated by means of a number of practitioner panels, held in different countries and with different practitioner communities (The TRESPASS Project, D4.2.2, 2016). The practitioner panels were run using provocations and stimulus material developed within the project, including tools and prototypes. Paper prototyping is a means of creating a paper version of a digital interface and inviting a participant group to engage with the paper prototype simulating the use of the digital interface. This has placed emphasis on taking paper prototypes to user groups to explore how they perceive risk through successive spheres: organisational, physical, digital and social. The importance of the spheres is to steer participants toward awareness of four significantly different views of the same issue, for example the differences between the views that security is about compliance to protocol (organisational), locking the office door (physical), changing passwords frequently (digital) or trusting a colleague with sensitive information (social).

A mapping kit was developed for these sessions. The mapping kit was composed of:

- A map of a geographical location (in most cases a room)
- Icons for physical assets and people
- Icons representing boundaries
- Colouring pens
- Tape

In each session the same process was followed. The process steps were as follows:

- Introduce the TRESPASS project and the role of visualisation within the project.
- Present participants with a scenario and a mapping kit and explain how to use the mapping kit.
- Ask participants to identify the assets, the connections between the assets and the possible attack paths.
- Place a likelihood on the success of each attack (represented by an attack path).
- Rank the risks based on the likelihoods.

The results were recorded through photography, note-taking and the collection of the completed paper prototyping. All panelists emphasised that risk impact should be the focus of visualisations as this is crucial to an analyst's activities. This is a good example of where visualisation provides a useful feedback to the modelling activities in TRE$_S$PASS. The approach is therefore to evaluate the inputs to the visualisation process to see where impact does feature and to also identify where it might feature. Where impact is addressed, the visualisations ensure that it is highlighted and where it is not addressed this gap is communicated to the TRE$_S$PASS technical co-ordinators.

The paper prototyping and discussion work with the practitioners highlighted that most practitioners seem to start with the assets and assess what they need to do to protect these. As a result, the Attack Navigator Map uses maps that are asset-centric so that practitioners have a degree of familiarity with the visual layout.

Processes were also identified as another form of asset. Process hierarchies were specified as a way to describe the way the different departments, roles and actors rely on each other in doing their job within the company. This potentially provides a way of visualising the interaction between the attack pattern libraries and the physical, virtual and social assets.

The feedback from these panels has provided valuable information for process development across the project.

6.2. Risk visualisations

Our approach to the critical visualisation of socio-technical dimensions lends itself to the iterative development of the understanding of an analyst faced with complex socio-technical scenarios. It is envisaged that over time the analyst will be assisted by the tools we propose, as they refine their representations and their interpretations of these, giving them greater detail where needed, and developing the metrics associated with the scenarios as they uncover further elements and add them to their models. Visualisation, and through this, interaction with the data, has therefore been carefully designed so as to allow the user to query and refine their information. This has been achieved via a range of activities and visualisation iterations, which lead to an informed interpretation and decision-making process.

Visualisations are to be made readily available to the user, allowing them to obtain an improved overview. Design affordances have been added to the interface that will allow the user to demarcate the various spheres in an operationally convenient way, as they add information about the social dimension, including actors, as distinct from the infrastructural or technical dimensions. Within this, social engineering steps may be visualised within a relational context of service provision.

Deliverable D4.2.1 (The TRE$_S$PASS Project, D4.2.1, 2014) presents a number of possible approaches to visualising attack trees, showing different ways to represent relationships and layering, according to the environment being represented. Representations of key

aspects such as difficulty, time and probability are shown in different ways, including increasing line width, colour and intensity. Additionally, the visualisations present scenarios as attack trees in many different forms and from many different perspectives, highlighting the features which are likely to be of most interest in a particular situation.

6.3. TRESPASS Attack Navigator Map

The attack navigator map, a central tool within the attack navigator interface environment, as described in Deliverables D6.2.2 and D6.3.1 (The TRESPASS Project, D6.2.2, 2015; The TRESPASS Project, D6.3.1, 2015), is a vital visualisation element of the project: This is a tool that predicts and prioritises attack scenarios based on a model of the system or organisation concerned. It can also be used to judge the effect of countermeasures, by re-running the analysis with an adapted model. The model takes the form of a navigator map and a set of attacker profiles. The navigator map represents the system cartographically, displaying connections between the elements as potential steps that an attacker could take. These steps are annotated with relevant variables such as difficulty and cost. The attacker profile collects relevant characteristics of an attacker, such as skill, resources, motivations / goals, and initial access. The latter can be thought of as a starting point on the map. For a combination of a map and a profile, the system will calculate routes for the attacker across the map that provides utility to the attacker. Typically, this will involve gaining access to certain assets and compromising their confidentiality, integrity or availability, which may cause damage to the organisation. The routes with the highest utility for the attacker constitute the highest risk with respect to the selected attacker profile. Multiple profiles can be combined to provide an overall risk picture.

6.4. Conclusions

A wide range of visualisation techniques have already been developed within the TRESPASS project. These have been applied in many different areas of the project, including the interface and as a means of illustrating the features of the analysis techniques. The approaches being developed have been validated through practitioner workshops.

7. TRESPASS Processes in Case Studies

The TRESPASS case studies have provided an important structure within which to guide the development of processes within the project. They provide realistic and varied input to data, modelling, analysis and visualisation processes, as well as providing an environment in which to evaluate the tools being developed within the technical work packages. The emphasis has been greater on different case studies at different stages in the project. In the first year, the IPTV case study provided a useful environment for investigating the complex relationship between technical and social risks. The Telco case study has tended to have a more commercial focus, while the Cloud case study has dealt with the technical complexity of virtualised platforms. The ATM case study has started later in the project and brought a greater focus to physical and geographical aspects of risk.

7.1. IPTV Case Study

7.1.1. IPTV overview

In this section we describe the "IPTV case study", that has been used as a running example throughout the project [1]. The technical details as well as the people and companies involved are confidential and we therefore present an anonymised and slightly redacted version of the original case study. However, all the important features have been retained and the processes are fundamentally the same as in the original case study.

The case study concerns a system supporting primarily elderly and disabled people in performing online payments and managing their own money from their home. With the target demographic in mind, the system should be integrated into an existing device that is familiar and easy to use for the intended user groups, namely the television set. In practice this is accomplished by hooking up a small, dedicated computer to the TV and an enhanced remote control with a built-in card reader for authentication as illustrated in Fig. 7.1. In this case study there are many different security aspects that may be considered: from the strictly technical, such as how information is protected while stored or transmitted, to the socio-technical, covering security issues arising from the use of and interaction with the technology. Within the project, we have explored the socio-technical features of the case study, as a means of validating the methods and tools which are being developed. In particular this case study has provided a context in which to explore the many possible approaches to handling social data.

[1] Here IPTV refers to television service(s) over the IP protocol.

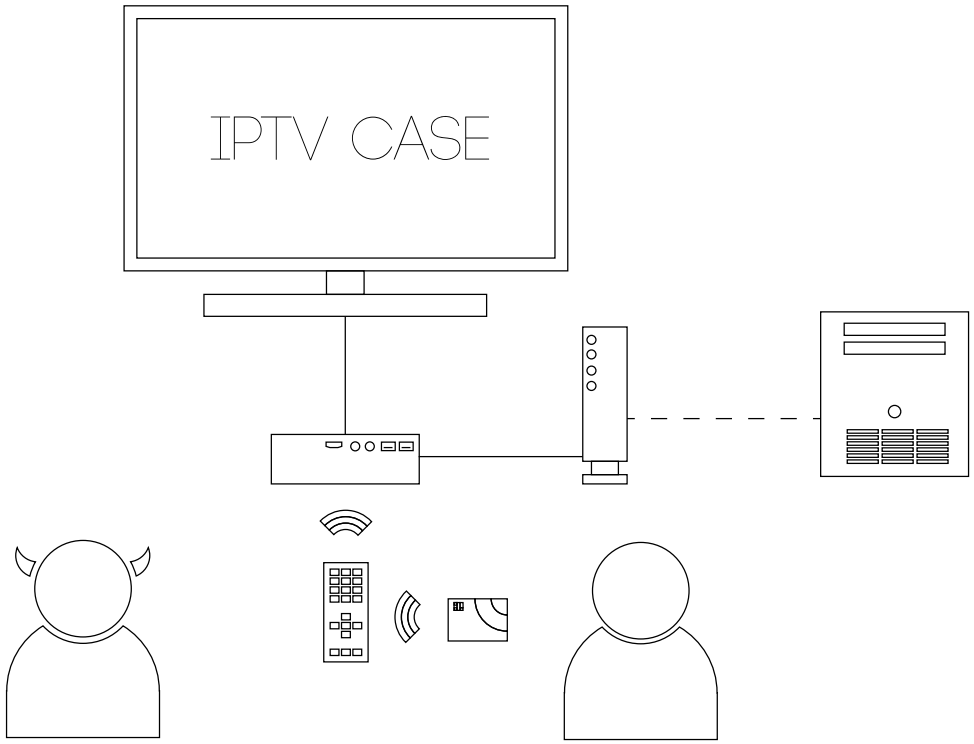


Figure 7.1.: IPTV case study

7.1.1.1. Normal Usage and Context Assumptions

Fig. 7.1 shows an overview of the IPTV case study. There are two primary actors: the attacker (represented by a devil in the figure) and the victim (the IPTV owner/user). Under normal operation, the user would first open a session on the IPTV, using a standard password based authentication scheme. From this session, the user can then use different services, e.g., pay a bill or transfer money, by using a payment card with concomitant PIN code. The payment card is read by a card reader built into the IPTV remote control on which the PIN code is also entered.

The design is intended to be simple and to offer the means for people of all ages and abilities to be able to access the services they require. It is intended to complement other means of delivery, not to replace them. In particular, it offers the opportunity for people who are not familiar or comfortable with mobile technology to receive the benefits of the ever-increasing range of mobile services in their homes via their television screen (Egelman, Brush, & Inkpen, 2008) . Although this system could offer great convenience, it also has the potential to expose the account holder to significant social risks, particularly those stemming from the involvement of both professional carers and family members. These carers could be considered as knowledge insiders, with the potential to act as malicious insiders.

We have made a number of assumptions about the context for the case study:

1. The card-holder has a functional IPTV in his/her house prior to the attack.
2. The IPTV security configuration ensures security for the communication of data between the different physical devices.
3. One Internet Service Provider (ISP) is used for all Internet access.
4. The source code of the software of the IPTV system is not freely available.
5. Firmware updates are not cryptographically encoded.
6. The IPTV set-top box uses a standard API.
7. The user can log on and off the IPTV system at will.

While these assumptions help delineate the scope of the case study, they are not critical and can be relaxed or modified to better capture a specific system.

7.1.2. IPTV process evaluation

Due to the comparative simplicity of the technical architecture involved in the IPTV case study, it formed the focus of much of the early modelling work in the project. It offered a convenient reference point for testing different approaches and evaluating their relative strengths. Apart from describing a simple architecture, without the added complexity associated with virtualisation in the cloud case study, the IPTV case has provided a rich social context in which to develop sophisticated processes for gathering and analysing social and organisational data.

The processes which have been applied to the IPTV case study are described here in line with the usual steps in the TRE$_S$PASS process: data collection, modelling, analysis and visualisation. In practice these techniques were applied and tested at dfferent times, frequently in parallel and sometimes at different stages in the project. The aim of this process was to gain further insight either into the case study or into the technicques being applied. For example, the CHYP SRA was applied at the very first stage in order to test the SRA process and to help partners gain familiarity with the case study. While valuable lessons were learned from this process, the CHYP SRA in that form is not part of the usual TRE$_S$PASS process. Next a hand-crafted attack tree was created during a workshop of project participants, in order to provide a more substantial basis from which to approach the case study. In parallel work was being done on modelling the case study with the TRE$_S$PASS model, which culminated in the ability to automatically generate attack trees from the model.

7.1.2.1. CHYP SRA

An extensive risk assessment using CHYP's Structured Risk Analysis (SRA) methodology (McEvoy & Whitcombe, 2002) was undertaken by UT early in the project, as described previously in deliverable D7.1.1 (The TRE$_S$PASS Project, D7.1.1, 2013). The purpose of

this initial work on the IPTV case study was to evaluate the SRA process in the context of TRESPASS and also for other partners within the project to gain familiarity with the case study. The same process was undertaken by CHYP risk practitioners shortly after the UT assessment. Based on the information provided regarding the high-level architecture of the system, first a logical data model of the information entities managed by the system was drawn up using a structured systems analysis approach. This data model was used to identify threats relating to each of the entities identified. Then a basic physical system model was used to derive possible vulnerabilities. These threat and vulnerability catalogues were used to cross-examine the relationship between system components and information entities, and attack profit and probabilities were estimated in order to generate a risk catalogue. This final table was used to prioritise possible countermeasures. While the areas of concern highlighted by each group were similar, there were significant differences in the priorities given to different risks, according to the practitioners' areas of expertise.

One important outcome of the first UT SRA process was that the system as proposed was considered to be inherently too insecure to go ahead without further efforts to secure particular aspects of the implementation. This concern was shared with our project partners and led to discussions of how to introduce appropriate security measures. This led into the risk evaluation workshops, led by RHUL.

7.1.2.2. Data collection

Because the IPTV scenario is relatively wide-ranging, a number of different approaches to data gathering have been adopted in this case study. Until M12, the focus was on the technical aspects of the system and the wider social context. From M12 onwards, different aspects of social data extraction have been explored in greater depth. The TRESPASS project uses both a hard and soft systems approach to modelling human behaviour. A hard systems approach creates models from bounded entities (assets) whereas a soft systems approach creates models from the perspective of human conceptualisations and perspectives. As a result a hard systems approach to modelling will not include the facility to view the model from different social and psychological perspectives whereas a soft systems model does.

The data collection process for the IPTV case study to date has included soft systems approaches applied at the organisational level and more statistical approaches applied at the level of the individual.

A central aspect of the case study is the system user's vulnerability to fraud. Research is ongoing to investigate whether victimization of four types of online fraud are related to Routine Activities indicators and socio-demographic characteristics in the same way that these are related in more traditional crime. The four online fraud types of interest in this scenario are identity theft, receiving fraudulent emails, accounts being hacked and online banking fraud. The figures regarding victimisation are derived from the Eurobarometer crime survey (European-Commission, 2014) and the main findings can be summarized as follows: users who spend the most time online, are also most often victimized for each

of the four categories of victimization. For fraudulent emails and hacking, the relationship with frequency of use and victimization is linear: the more users are online, the higher the victimization rates. For identity theft and online banking fraud the relationship is less clear. Further details of this research are included in Deliverable D7.4.1 (The TRESPASS Project, D7.4.1, 2014).

7.1.2.3. LEGO modelling

From discussions with our case study partner, a set of security concerns emerged relating to untrusted behaviours that might pose risks to the system, both of users and competitors. From here a visualisation was developed to encompass the range of concerns, including both technical and social, expressed in relation to the proposed innovation. The analysis of the proposed service was achieved through the use of a specific combination of visual methods. This was aimed at gathering hard-to-reach data, while simultaneously finding the appropriate forms with which to visually represent the data-paths and the flow of data along these.

LEGO was deployed with the developers of the planned IPTV home payment service to co-construct a rich multi-perspectival and layered picture of data-sharing as a part of the service. Participants were asked to model (using the colours and language of ArchiMate) the central actors (yellow bricks), infrastructure (green bricks), data (blue bricks), and locations (pink tiles). The weighting and positioning of each element was collectively agreed upon. Patterns of data-flow were observed across spaces of trust, and the model was enlivened by the participants' use of LEGO avatars to represent the central actors and the control strengths of selected points along data-paths. In particular, the second session was treated as an opportunity for the group to reflect upon and remodel the weaker parts of the service design, which were enhanced with further bricks to that effect.

This approach to gathering data while modelling the wider landscape in a participatory environment has now formed the basis of the 'Stage Zero' modelling within the project.

7.1.2.4. TRESPASS model

A model of the IPTV case study was developed as input to WP1 deliverable D1.3.1 (The TRESPASS Project, D1.3.1, 2013). This model highlighted the relationships between the different actors in the case and the potential relationships and actions involved in the case study. The modelling focused on the overall aim, to prevent the cardholder from being parted from her money. The model and associated tools provided a useful structure in which to explore the socio-technical risks presented by the IPTV case study. Subsequent work on policies and attack generation in WP1 and WP3 have provided an increased level of understanding in this respect.

The TRESPASS model combines the infrastructure view of ExASyM (Probst & Hansen, 2008) and the actor view of Portunes (Dimkov, Pieters, & Hartel, 2010), introducing a modular structure with the goal of making it straightforward to model human behaviour

and to include it in the analysis. This modelling approach has proved to be well suited to the range of social and technical attacks encountered within the IPTV case study. In particular, the scope for representing relationships between different actors is a significant strength.

7.1.2.5. Attack trees and analysis techniques

In order to investigate the case study further, the risks associated with the IPTV case study were mapped to create an attack tree. This provided a very comprehensive outline of the types of attack to which the system might be vulnerable. The features of this tree are described in more detail in deliverable D3.3.1 (The TRE$_S$PASS Project, D3.3.1, 2013).

The creation of the attack tree was also valuable in identifying the processes which reappeared frequently within the same tree. In some instances, these processes could potentially have been applicable across both IPTV and Cloud case studies (for example compromise of network equipment). This provided the opportunity to develop processes for handling frequently occurring situations and to explore the strengths and weaknesses of the existing modelling approach. In addition, the frequently repeating nature of branches within the tree highlighted the need for a mechanism to store and retrieve frequently occurring elements of a tree. It was in this context that the proposed implementation of pattern libraries was discussed.

A number of different analysis techniques were applied to the data collected in the large scale attack tree describing the IPTV case. These included the use of timed automata by AAU and the ATree Evaluator tool developed by DTU. These permitted the computation and ranking of optimal attack paths for different quantitative attributes such as time, budget, skill or damage. The use of the Pareto tool permitted the effects of different trade-offs between potentially competing attributes to be evaluated.

7.1.2.6. Visualisations

Visualisations were found to be a very successful part of the engagement with our case study partner. From the original Archimate modelling, which was very well received, to the later stages of the LEGO model development, the use of visualisations helped to achieve a greater level of understanding within the group. It also helped to foster a greater sense of ownership within the organisation, who continued to update the emerging LEGO model even between workshop sessions. The visualisations developed during these sessions are presented in detail in deliverable D4.3.1 (The TRE$_S$PASS Project, D4.3.1, 2014).

7.2. Cloud Case Study

7.2.1. Cloud overview

Cloud computing has gained remarkable popularity in recent years due to the economic and technical advantages of this new way of delivering computing resources. Customers benefit from rapid provisioning and seemingly infinite scalability, while only being charged on a pay-per-use basis. Computing resources can be provided on different abstraction layers (Mell & Grance, 2009b), where the lowest one provides basic resources (servers, network, and storage), and higher ones provide applications to the end-users (e.g., Google's GMail, Salesforce.com). In this case-study we are focusing on the lowest abstraction layer, that is *Infrastructure-as-a-Service* or *Infrastructure Clouds*, since it is the most generic layer and higher ones often build upon this layer.

Although the benefits of cloud computing are evident and users demand cloud services, security is a major inhibitor (Mell & Grance, 2009a). An analysis of risks and threats in cloud computing has been conducted in (Cloud Security Alliance, 2010) and (ENISA, 2009). In particular, both reports agree that insider attacks and malicious insiders are a major risk and are among the top 10 threats. The risk is amplified due to the disappearance of physical boundaries that makes it very challenging to define a security perimeter that divides insiders from outsiders (Hay, Nance, & Bishop, 2011; Pieters, 2011).

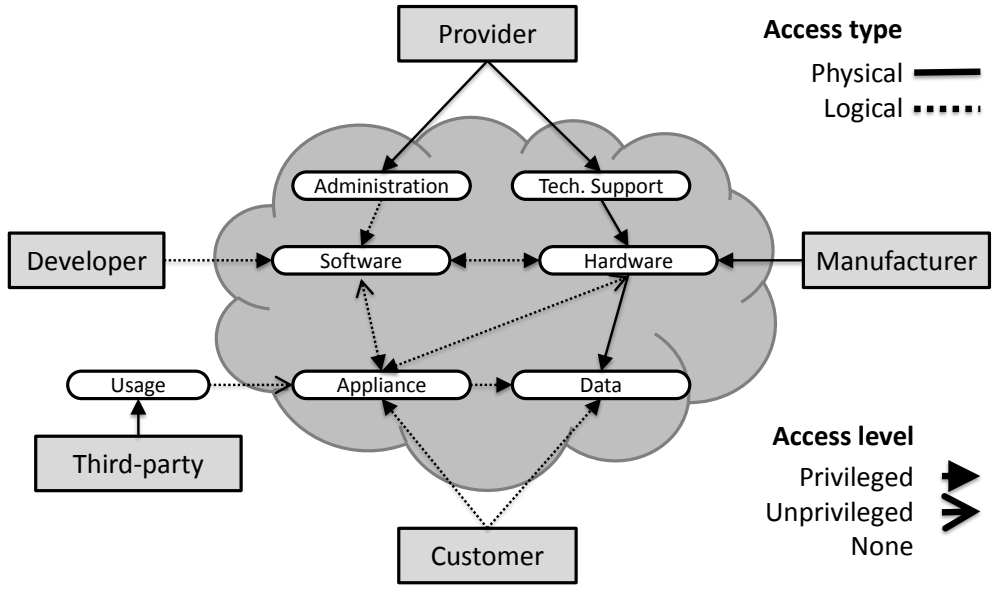


Figure 7.2.: Overview of entities and components in an infrastructure cloud model.

Figure 7.2 provides an overview of the entities and components involved in a model of an infrastructure cloud. As compared to a typical IT department within an organisation, the provider of the cloud services in such a shared infrastructure cloud is a new additional and powerful party. The multi-tenancy (different customers, including competing organisations,

using the same cloud services) lead also to an unprecedented sharing of computing and infrastructure resources. The cloud provider therefore becomes an important additional factor (and risk), as it has full physical and logical access to all resources across the different consumers. In addition, the infrastructure itself is dynamic and flexible in response to the requirements of the customers.

If risk assessment in complex technical infrastructures is already difficult, it is even more complicated when human factors and physical infrastructure are added to the setup, which are correspondingly often ignored (Probst & Hunker, 2009).

The special interest to investigate this scenario within the TRE$_S$PASS project is therefore to develop models and processes that support risk assessment in complex organisations *including* human factors and physical infrastructure. The goal of this support is to simplify the identification of possible attacks and to provide qualified assessment and ranking of attacks based on factors such as the expected impact.

For cloud infrastructures, the TRESPASS model distinguishes components at a level of abstraction that corresponds well to security-relevant control points in these domains, enabling the discovery and analysis of potential attacks that exploit their connectivity. Using the model, one can formalise typical components in cloud infrastructures and their interrelationships. These include network components like switches, routers, firewalls; virtual and physical servers; actors, including administrators, users, and attackers; location details that represent rooms, doors, and other physical consideration. Because these component models show how actions on one element influence other elements, they can be combined with the connectivity relations to form an implicit search-space of all possible activity paths in the system.

7.2.2. Cloud process evaluation

Progress for the cloud case study is driven forward by the definition of a simple cloud scenario with a physical setup, definition of a virtualised infrastructure and a set of personas following a common storyline. An overview of this scenario is shown in Fig. 7.3.

For demonstration purposes, a physical, portable demonstrator, nicknamed *Cloud-in-a-Box*, has been set up. Its physical setup is comprised of three small-form-factor PCs (Intel NUCs), connected by a small switch as network interconnect (see Fig. 7.4 for a picture).

These three PCs are running VMware ESXi 6.0 as the operating system, centrally managed by a vSphere management server implemented as a virtual machine sitting on top of one of the ESXi servers. As a small cloud setup, we implemented a virtualised environment with multiple department resources, partitioned network and distributed storage. Also, user ids and access rights are configured to follow the scenario personas and to be used in the demonstrator for the cloud case as base for testing and presenting the full TRE$_S$PASS processes.

A data extraction tool was built allowing the querying of the elements of the cloud and exporting of the data. In the envisioned data flow of the overall TRE$_S$PASS process this

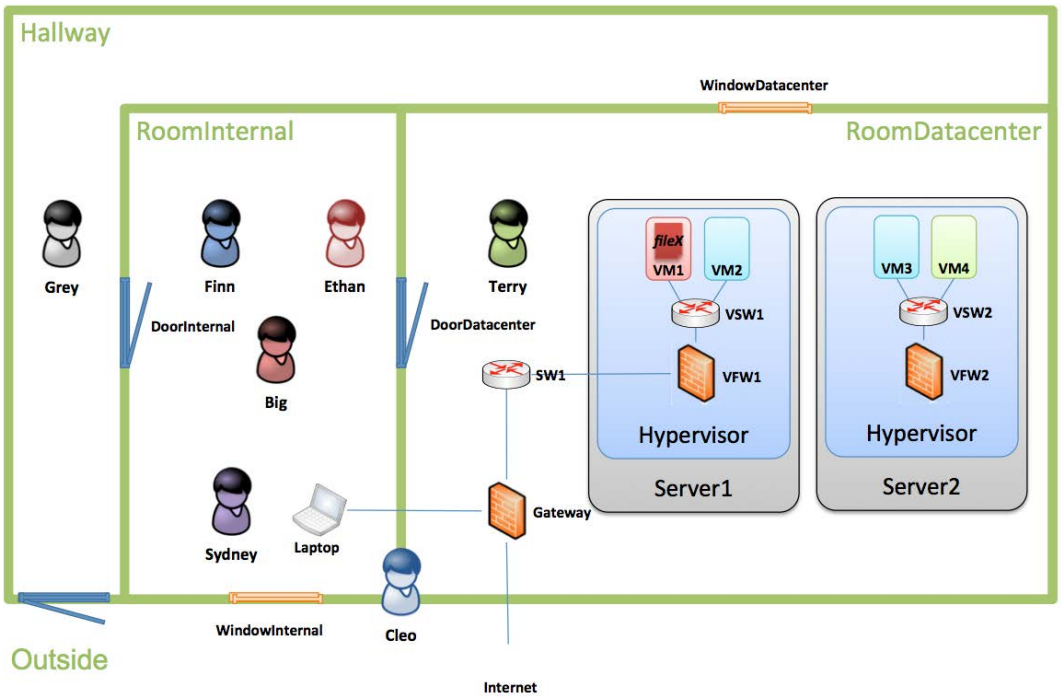


Figure 7.3.: Overview of the scenario implemented in the *Cloud-in-a-Box* demonstrator and used in this demonstration.

Figure 7.4.: Physical setup of the *Cloud-in-a-Box* that is used as the cloud infrastructure in this demonstration.

data extract containing the details of the virtualised infrastructure will be imported during the building of the complete model with the *Attack Navigator Map* (ANM). Through this, the elements of the cloud infrastructure are 'grounded' in the geospatial setup of the floor

plan on the one hand and its abstract user IDs as defined in the cloud setup connected to real-world actors (personas) on the other hand.

For a better understanding and overview of the extracted data a web-based viewer was also implemented, showing the interrelations of the cloud entities. The focus on this graphical representation is to allow viewing of the access control rights of various user IDs. Fig. 7.5 shows a screenshot when the user *Terry* is selected showing that Terry has only *ReadOnly* access to the `ProductionCluster`, but has the role *Network administrator* on the `TestCluster`.

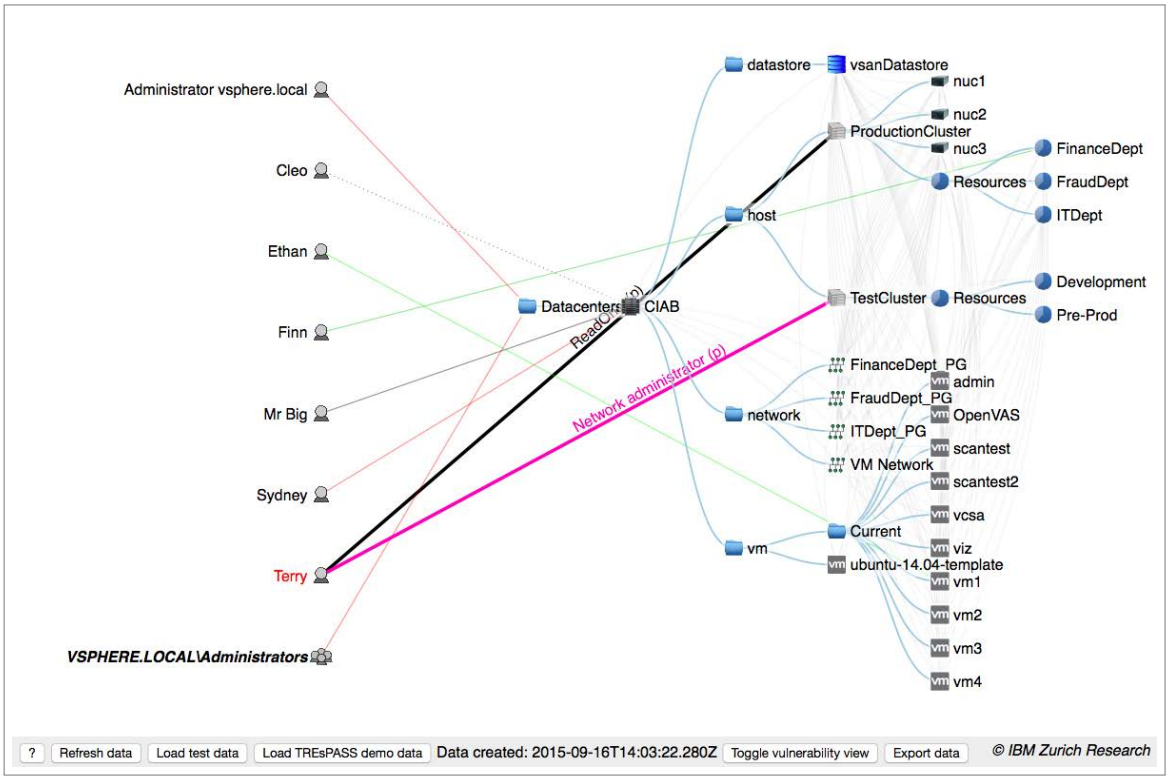


Figure 7.5.: Web-based view after selecting the user *Terry*, highlighting his access control rights.

For details about these tools see Prototype D2.2.2 (The TRE$_S$PASS Project, D2.2.2, 2015).

The generated extract defining the cloud infrastructure has to be manually merged with information about the rooms and the actors. Such a manually merged file has been used successfully as input for Treemaker as described in deliverable D3.4.1 (The TRE$_S$PASS Project, D3.4.1, 2014), but further adaptations have to be done to create attack trees for upstream use in the attack pattern library (APL) and analysis tools.

Using and discussing this cloud case scenario, we learned that further work is required on the way that concrete entities in the cyber infrastructure are to be handled and considered

in the following analysis: for example, a virtual machine will usually be in the memory of the corresponding host system during run-time only. The files of the virtual machine will rather persist on the corresponding datastore. Therefore, the link of a virtual machine to a host system means a rather different thing for use by attackers than the link to a datastore. I.e. stealing a host system alone will not allow access to the virtual machine.

Some form of this real-world understanding is therefore required to correctly investigate further attack steps during the backtracking of Treemaker. This mechanism is addressed in Deliverable 5.3.3 as traceability (The TRE$_S$PASS Project, D5.3.3, 2016).

7.3. Telco Case Study

7.3.1. Telco overview

Telecommunications products and services are functional in a complex environment involving a multitude of different interconnected networks, service providers and network operators with opposed financial interests acting in highly competitive markets offering complex products and services.

Due to the market structure described above, new customers are not easily available, but normally need to be lured away from competitors. As the providers try to escape the pricing pressure resulting from replaceability of the communication goods, e.g. new tariffs more and more become a mixture of free (flat rate) components being heavily advertised and, less noticeable, much more expensive components for compensation and revenue generation. New products need to be launched under significant time pressure resulting from strong competition in the market, leaving little time and space to account for potential misuse of the product. Often the misuse resulting from product design flaws is a learning process which takes place once the respective product or service has been launched into the market.

The above tendency, in contrast, encourages cherry-picking among customers of Telco companies. This is especially true for so-called knowledge insiders that know the market very well, trying to make as much use of (or monetary gain from) the products offered as possible.

We define Telecom misuse as the contracting or consumption of telecommunication services in a manner that is not in line with the service provider's expectations. Fraud is then any instance of misuse as previously defined with the explicit goal of obtaining financial rewards.

An example of the sort of fraud addressed by the project involves using insider knowledge to exploit telecom service provision structures and/or their respective terms and conditions in order to commit Revenue Share Fraud (RSF). This usually implies setting up a revenue sharing agreement with one provider, and a beneficial (usually flat-rate or unlimited) subscription with another, and then calling yourself (Figure 7.7). The fraudster can then generate income by triggering the payment of termination fees from one provider to

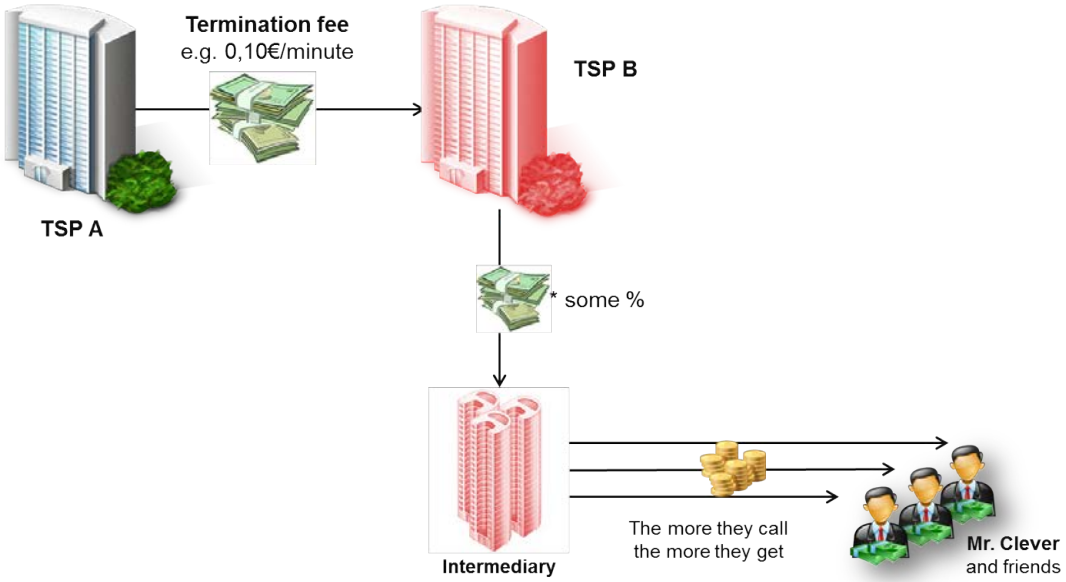


Figure 7.6.: Fraud Scenario: Revenue Sharing Fraud (common set-up)

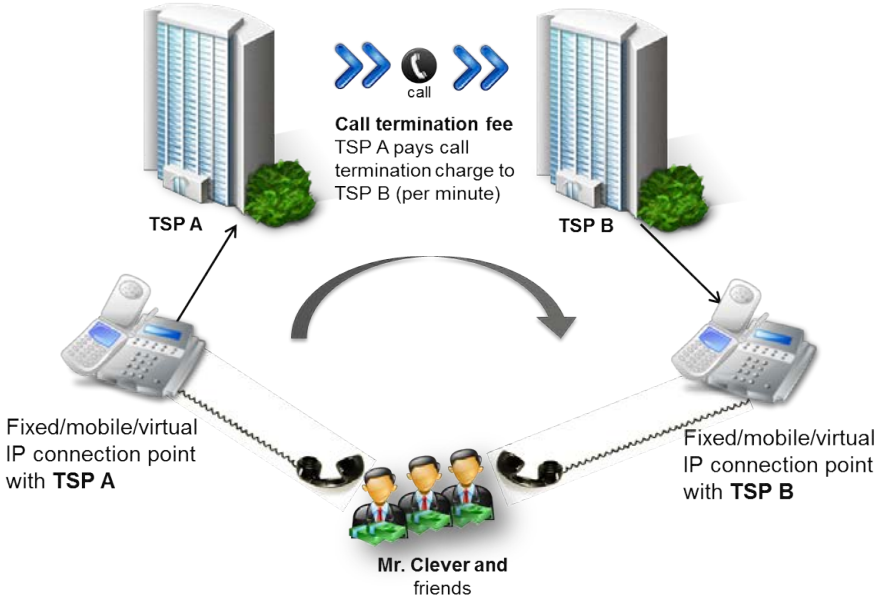


Figure 7.7.: Fraud scenario: Revenue sharing fraud (money flow)

the other, (refer to Figure 7.6). Depending on the scale of the operation and the detection capabilities of the provider, fraudsters can pull in up to several million dollars over a weekend using such schemes (Baker, 2012). Other examples include the false pretence of being willing and able to pay for calls (Figure 7.8).

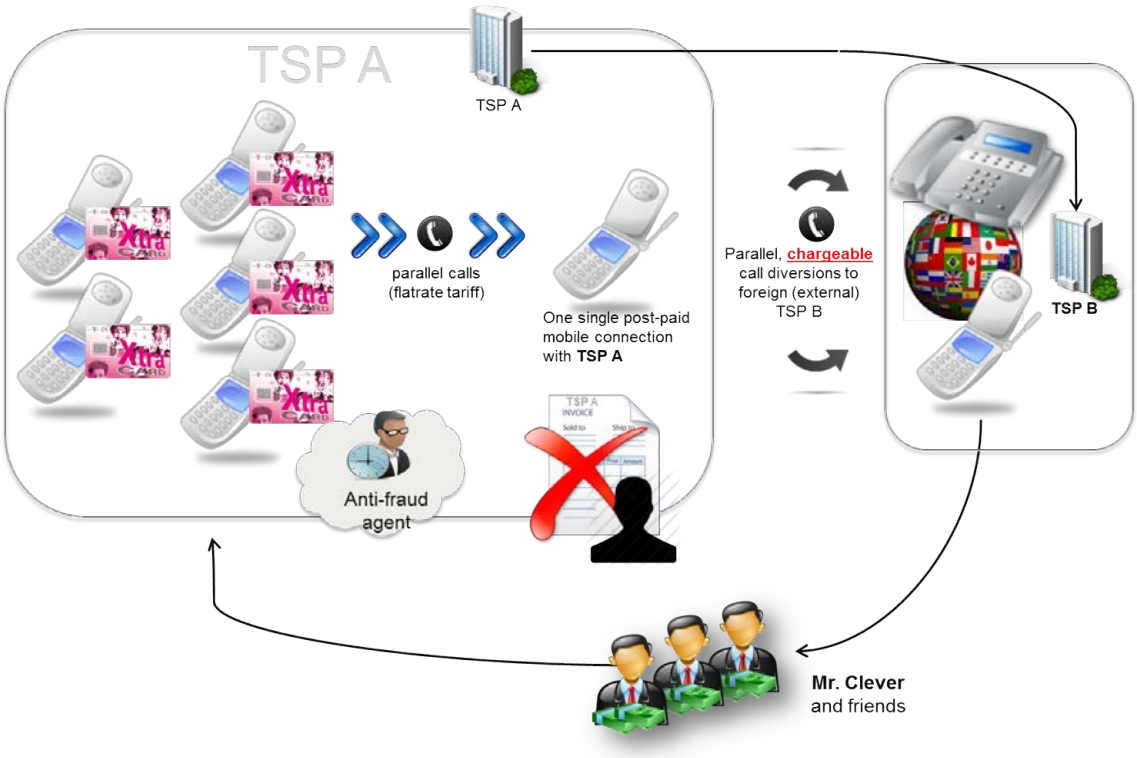


Figure 7.8.: Fraud scenario: Fraud involving the false pretence of being willing and able to pay.

7.3.2. Telco process evaluation

In order to quickly estimate the potential for loss of a given service package, as well as identify thresholds useful in drafting “Fair-use policies” and fraud detection heuristics to limit this loss, traditional heavy-weight Risk Assessment paradigms are of little use. In a previous structured literature survey (The TRESPASS Project, D5.2.1, 2014), we observed that established modelling techniques commonly used in Risk Assessment are focused on modelling the technical or socio-technical aspects of an (information) system and are mostly concerned with risks affecting confidentiality, integrity or availability. Furthermore, the methods employed are likely to take days or weeks to thoroughly apply (Ionita, Hartel, Pieters, & Wieringa, 2013).

Indeed, initial modelling efforts using the TRESPASS language revealed similar limitations of the language. Many of the concepts required by the TRESPASS language were superfluous when modelling telecom fraud. An investigation of several known telecom fraud scenarios revealed that, contrary to more technical attacks, they are best described in terms of value exchanges (such as money or services) amongst profit/loss responsible actors.(The TRESPASS Project, D7.3.1, 2014).

To handle the very specific requirements of the Telco case study, namely the focus on value transfers instead of the socio-technical architecture, an alternative modelling and analysis approach was developed. This approach is designed to also cover (most of) the dynamic features required. The e3fraud modelling language and tool are described in Section 2.3.2 of D1.3.2 (The TRESPASS Project, D1.3.2, 2015), while the e3fraud risk assessment process is outlined earlier in this document. The e3fraud methodology is also presented in detail in (Ionita, Wieringa, Wolos, Gordijn, & Pieters, 2015).

7.4. ATM Case Study

7.4.1. ATM overview

ATM machines are composed of a money safe and a computer that controls the ATM's devices (screen, keyboard, printer, network interfaces, money safe mechanisms, etc.) through software programs. Most ATM computers are composed of outdated hardware running legacy operating systems (i.e. Windows NT), unsupported by the vendors and by the anti-virus providers. The installation of the ATM machines varies as there are ATMs installed in well protected bank branches, while others are deployed in the street and not even embedded in a wall.

ATM attacks are common and include classic physical attacks and emerging digital attacks. Examples include:

- Physical attacks where the attacker physically steals the ATM to open the safe and take the money.
- Digital attacks where the attacker installs a malware agent into the operating system to take control of the devices including the ability to withdraw money from the safe through the device' s interfaces.

In order to perform a proper risk assessment of the ATM network as a whole, four types of data must be considered and combined into a single unified model, suitable to be processed on a consolidated data schema:

- Technical data (about the machines);
- Social data (about socio-demographic/economic variables);
- Environmental data (about the territory);
- Historical data (about past occurrences).

Attacks do not happen randomly or equally distributed over the territory. Machines installed in certain places are more likely to be attacked than others, thus hotspots exist. This case study therefore adds a geographical dimension to the project.

The ATM case study builds an analysis process at the macro-level based on data from the ATM network and surrounding area, and outputs the vulnerability level of each ATM. More specifically, this case study illustrates how to perform an analysis at a macro level

for identifying priority areas in need of detailed analysis using the TRESPASS model and tools.

7.4.2. ATM process evaluation

In the case of the ATM case study, there is a need to extend the model with additional capabilities, specifically regarding the geographical nature of the case study. This case study involves multiple levels of analysis:

The first level evaluates the statistical likelihood of attacks in a particular area (i.e. some locations within a particular area may be subject to higher levels of crime than others: levels of crime are not generally uniform across a geographic area). In order to infer this first level, a range of data may be used: demographic, socio-economic, environmental, based on historical events. This level of statistical evaluation provides a high level overview, which is of considerable valuable to the authorities. The police may use this kind of information to allocate defensive resources appropriately, concentrating patrols in areas where the types of crime concerned are particularly prevalent.

At a more detailed level, a TRESPASS analysis may be undertaken, providing a precise level of information that takes into consideration the very specific context where the machines are deployed (building, doors, walls, cameras, technical detail of the machines, etc.). This level of information is of considerable value to the owners of the infrastructure, as it provides information on ways in which resources may best be deployed in the specific ATM locations.

These two levels of analysis are complementary, in that the detailed level must take into consideration the statistical level of evaluation: two identical machines, deployed in two identical buildings, such as petrol stations built to a standard plan (detailed level), can have completely different risk profiles (likelihood of being attacked), where one is located in an area which has an historically high level of social upheaval, while the other is not. Each approach is tailored towards different end users: from police authorities (statistical level), to infrastructure owners (detailed level). For that purpose, we'll start by extending the TRESPASS base model with geographical features, since the statistical level relies on it. Other extensions to the base model are, of course, possible. The most important message is that, in order to allow different scenarios of application, the TRESPASS base model must be flexible enough to accommodate extensions to the data model and processes.

To date, the project has demonstrated the data gathering and consolidation process in relation to this case study, including the generation of a statistical view (layer) of the area where ATM machines are deployed, with an estimation of the likelihood of each ATM being attacked, per attacker profile and type of attack. From this, we have found that a statistical layer identifying the likelihood of ATM attacks across an area and the specific hotspots, is an important tool for authorities to plan defensive resources (i.e. patrols), since it presents a global view of the ATM infrastructure across that area. Beyond the statistical layer, a detailed TRESPASS analysis can be included to complement the overall risk analysis, using the TRESPASS model and extended analysis based on attack trees, which includes

information about the location where each machine is deployed (room, facilities, walls, doors, etc.) as well as other attributes.

Crime analysis involves the collection and analysis of data related to a criminal incident (attack), offender (attacker), and victim (target). In this context, a TRE$_S$PASS data management process must understand and be aware of relevant aspects related to crime analysis data, more precisely being able to support the identification and generation of the information needed to assist the analysis and decision processes and the deployment of efficient countermeasures to prevent and reduce the likelihood of criminal activity. Over the coming year, the project plans to develop the existing statistical view and extend it with the complementary TRE$_S$PASS detailed analysis process.

7.5. Conclusions

In this chapter we have described the application of the elements of the TRE$_S$PASS process to different case studies according to their individual needs and characteristics. While there are many common requirements between the case studies, especially in the core social and technical areas, their individuality has also permitted a number of additional processes to be incorporated into the TRE$_S$PASS approach. For example, the adoption of the e3value method for the Telco case study has provided insights into handling commercial and contractual aspects of risk which are less well suited to a more traditional architectural approach.

While the usual steps in the TRE$_S$PASS process might include social and technical data gathering, followed by the creation of a TRE$_S$PASS model, which is used to generate attack trees for further analysis, this process is very adaptable according to the needs of the individual case studies. One thread that persists throughout is the value of visualisation in presenting the different elements to the end user.

Our experience in working with case study partners has shown that the TRE$_S$PASS tools and techniques are powerful enough to model and analyse a wide variety of different case studies. In addition, the results can be visualised in a way that is helpful for practitioners.

8. TRESPASS Processes in Relation to ISKE

ISKE (*ISKE*, 2013) is a baseline security system originating from the German BSI standard (*BSI*, 2013) and adopted for the Estonian environment. It is based on a huge catalogue of enterprise asset types, related threats and countermeasures which permit the user to map enterprise assets to ISKE asset types, assign the desired security levels for assets and generate a list of related countermeasures from the ISKE catalogue. The countermeasures need to be deployed in order to maintain and enforce the protected assets at the required security level. The ISKE system does not prioritise either threats or countermeasures and therefore does not provide analysts with a sufficiently detailed picture of the surrounding risk landscape. TRESPASS on the other hand aims at quantitative risk analysis, which allows for the prioritisation of risks and related countermeasures. Integrating ISKE with TRESPASS would make it possible to enhance ISKE and to produce a list of related countermeasures, sorted from the most critical to the least critical. Since ISKE is mandatory for public sector enterprises in Estonia, it is a worthwhile goal to integrate ISKE and the TRESPASS toolset. In the subsequent sections we outline the case study aimed at investigating the possibilities and limitations of integrating the TRESPASS analysis process into ISKE.

8.1. ISKE Baseline Security System

As was stated earlier, ISKE is a huge catalogue containing a finite set of predefined asset types each of which has a list of related threats, while each threat in turn is related to a set of security measures to mitigate it. We refer the reader to Fig. 8.1 for reference.

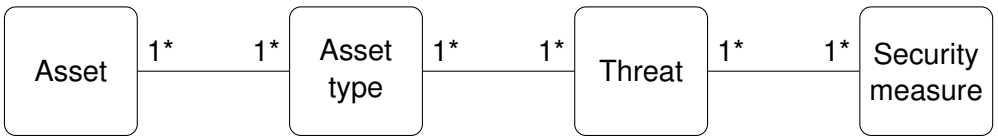


Figure 8.1.: Relations between the major data types in ISKE

The security measures are grouped by security levels – tuples containing three components: corresponding levels for asset *confidentiality*, *integrity* and *availability* which are defined in the ISKE standard as shown in Table 8.1.

In order to enforce and maintain the protection of an asset at a requested security level, the corresponding subset of security measures is selected from a bigger list of security measures aimed at mitigating threats for the considered asset as shown in Fig. 8.2.

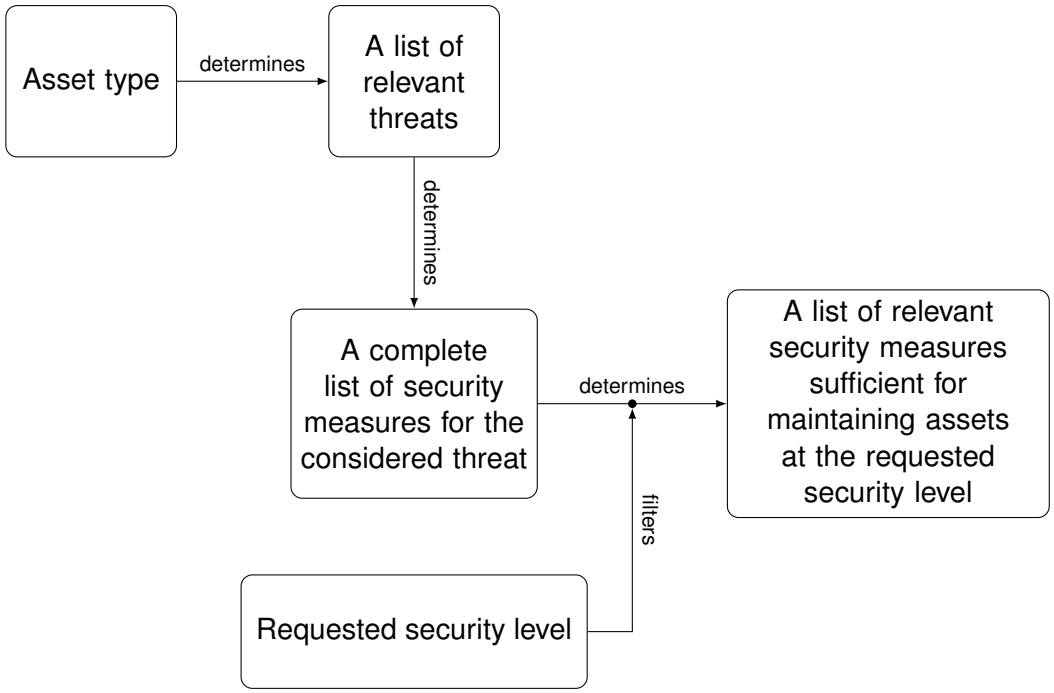


Figure 8.2.: Security measures are determined by *asset type* and the requested *security level*

The higher the requested security level, the longer the list of security measures and procedures that have to be in place to achieve and maintain the requested security level for the asset concerned. ISKE neither prioritises security measures nor threats and thus provides no hint as to how (e.g. in what order) to implement the security measures from the list.

From the end user perspective the ISKE workflow looks as follows:

1. The end user lists assets of the enterprise concerned.
2. Each of the assets is assigned with asset types from the ISKE catalogue. There is a 'many to many' correspondence between the real asset and its type in the ISKE catalogue, as there may be several real assets of the same asset type (consider for example dozens of PCs in the office premises) and likewise a single asset may correspond to several asset types (e.g. a personal PC may be assigned with the following asset types: client PC, PC connected to the internet, autonomous IT system).
3. Each of the assets is assigned with the desired or required security level. This level will influence the amount of security measures to implement and also most likely the amount of the corresponding security investment. The higher the security level is the

more security measures need to be implemented in order to enforce and maintain a given asset at a requested security level.

4. For each of the assets the user runs the following procedure:
   - For a given asset a list of related threats is obtained;
   - For each threat the user obtains a list of countermeasures;
   - Given a security class defined for an asset the user then selects the subset of security measures required for implementation to keep an asset at the requested security level.
5. The countermeasures required for implementation are collected for every asset and then incorporated into one single list of security measures.

Such a list of security measures is the result of a single iteration of the ISKE implementation process. Based on this list, the end user can calculate the required security investment and time required for implementation of the security measures. ISKE does not provide any suggestion as to which countermeasures are most urgent, nor which ones are less critical and may be postponed until the next iteration. The end user makes decisions on the relative importance of countermeasures based on expert knowledge and experience, as well as balancing these decisions with the cost of implementing them within the available budget. This judgement can be fairly subjective.

8.2. Integration – possibilities and limitations

In order to enhance ISKE with the capabilities of the TRE$_S$PASS approach, the first objective was to establish a mapping between the ISKE model and the TRE$_S$PASS model. The ISKE model is basically a set of assets and corresponding security levels, while the TRE$_S$PASS model is a set of assets, policies and processes. It can be seen that the only model entity which can be mapped to one another in both models is the asset entity. Other entities, such as policies and processes in TRE$_S$PASS as well as security levels in ISKE are types of model entities explicitly existing in either TRE$_S$PASS or ISKE but not in both.

On further investigation, it turned out that assets could not be mapped to one another in a straightforward way either. The complexity of mapping assets arises from the fact that ISKE and TRE$_S$PASS use a different level of granularity for modelling assets. ISKE does not model particular assets but only generic asset types, which is insufficient for the TRE$_S$PASS approach of modelling every single asset in the enterprise concerned. The assets in the TRE$_S$PASS model may be grouped and such an entity as *asset kind* exists in TRE$_S$PASS, however the set of possible kinds of assets is rather generic (e.g. item, location, data) and does not correspond to the asset types in ISKE. Assets in the TRE$_S$PASS model can be uniquely mapped to the asset types in ISKE only if it is possible to come up with some sort of generalisation rule – a heuristic which maps a particular asset to a corresponding asset type in ISKE based on the asset name and asset kind

parameters. Following the observations from several case studies it became clear that TRE$_S$PASS models enterprises at a greater level of granularity than ISKE does and thus the mapping from TRE$_S$PASS assets to the ISKE asset types is a rough attempt to classify items. The effect of it on the relevance and precision of the results has not been estimated yet, but it might be desirable to extend the set of ISKE asset types to increase the mapping variance.

The security level of an asset lies at the core of ISKE and without it ISKE analysis would be meaningless. Following existing ISKE practices, the practitioners assign assets with the required security levels. Such an expert estimation is mostly subjective and is a best effort. Unfortunately, the level of modelling granularity used in ISKE is insufficient for deriving the security levels in a an automated way. The TRE$_S$PASS model contains more entropy compared to the ISKE model and thus it might be possible to come up with some sort of a heuristics engine which would assign corresponding security levels to assets by analyzing the TRE$_S$PASS attack navigator map. The engine in its initial implementation could follow simplistic rules, such as:

- Mission-critical assets require elevated levels of protection and should be assigned maximum affordable security levels;
- Assets storing or handling sensitive or classified information should be assigned elevated security levels. The same is true for the communication channels over which sensitive information is transmitted;
- The baseline security levels for assets could be derived by analysing SLAs and supply chains. In this case these entities must be explicitly modeled in the TRE$_S$-PASS model;
- A simplistic procedure could prioritise assets according to their mission criticality and correspondingly elevate the security levels (compared to the baseline levels) taking available budgets into account.

The merits of TRE$_S$PASS are that assets would be assigned with the required security levels capable of providing a maximal level of protection. The security levels would be optimal for the considered infrastructure, taking into account the current state of the surrounding risk landscape. The capability of taking the current state of the surrounding risk landscape into consideration is a major advancement of the TRE$_S$PASS approach compared to ISKE which is rather static and for which the update cycle is quite lengthy. The TRE$_S$PASS toolset allows security levels to be adjusted dynamically in accordance with the changes in the surrounding risk landscape and thus provides a justified optimal level of protection at any given moment in time.

Prioritisation of security measures is another area where TRE$_S$PASS could provide an enhancement to ISKE. The drawback of ISKE is that it does not prioritise security measures. As budgets are limited, every security investment must be optimal and must be justified. Priority should be given to securing the most vulnerable and mission critical parts of the system in the first place. ISKE simply produces a list of security measures and gives no suggestion as to which of them should be implemented urgently, and which may be postponed until a later stage. Some of the TRE$_S$PASS tools (for instance, the Attack Tree

Analyzer) are capable of assessing if the system is secure enough, and if not what the most likely attack vector is. The resulting attack vector consists of a set of attack steps targeted against various assets. It might be possible to obtain the list of the vulnerable assets in the enterprise at a particular moment in time. Thus some sort of prioritisation may be achieved, where the security measures are prioritised in accordance with their exposure criteria: the ones which are most likely to be attacked and the ones which are less likely to be attacked. When making a decision on which security measures to implement, the security measures of the mission-critical components must be implemented first, followed by the most vulnerable components according to the TRE$_S$PASS analysis results.

8.3. Conclusions

The integration of TRE$_S$PASS and ISKE is possible in principle. However such an integration would require certain heuristic claims to be made. The integrated workflow from the end user perspective is shown in Table 8.2.

Table 8.1.: ISKE asset security levels

Confidentiality	
S0	public information: access is not restricted (read access for everyone, modification access is regulated by the corresponding integrity requirements)
S1	information for internal use: access is granted upon request and a legitimate justification for it
S2	secret information: access is granted only to certain groups of persons upon request and a legitimate justification for it
S3	top secret information: access is granted only to certain persons upon request and a legitimate justification for it
Integrity	
T0	source of information as well as its modification and destruction traceability is not important checks for information accuracy, completeness, checks if information is up-to-date are not necessary
T1	source of information as well as its modification and destruction must be traceable information must be inspected for accuracy, completeness, checks if information is up-to-date shall be conducted only on special occasions or as needed
T2	source of information as well as its modification and destruction must be traceable information must be inspected for accuracy, completeness, checks if information is up-to-date shall be conducted periodically
T3	source of information as well as its modification and destruction must be traceable information must be inspected for accuracy, completeness, checks if information is up-to-date shall be conducted in real time
Availability	
K0	reliability – irrelevant; performance – irrelevant
K1	reliability – 90% (total permissible outage a week ~one day); permissible increase in required response time during peak loads - hours (1 ÷10)
K2	reliability – 99% (total permissible outage a week ~2 hours); permissible increase in required response time during peak loads - minutes (1 ÷10)
K3	reliability – 99.9% (total permissible outage a week ~10 minutes); permissible increase in required response time during peak loads - seconds (1 ÷10)

Table 8.2.: TRE$_S$PASS & ISKE integrated workflow

End user	**The TRE$_S$PASS toolset**
Models the considered enterprise	No action
Runs the analysis	Analyses the navigator map model. The results is a Boolean value showing whether the infrastructure is secure enough or not. In case it is not, the most profitable attack vector (from the attacker view point) is also given.
Switches to the ISKE view	For each asset found on the navigator map: • Assigns an asset with corresponding ISKE asset type using heuristic generalisation rules • Assigns an asset with a security class using the heuristic reasoning engine • Identifies mission-critical assets using the heuristic reasoning engine • Queries the ISKE catalogue and generates a list of relevant threats • Queries the ISKE catalogue and generates the list of relevant security measures, required to keep an asset at the specified security level • Assigns priority levels to security measures based on the following considerations: - Top priority – *mission-critical vulnerable* assets - High priority – *mission-critical non-vulnerable* assets, *regular vulnerable assets* - Normal priority – *regular non-vulnerable assets* Joins the security measures obtained for every single asset into a single list of security measures and outputs a prioritised list of security measures as the result.

9. TRESPASS Process in Relation to CORAS

Here we continue the exercise of comparing the TRESPASS approach to risk assessment with other established methodologies. In this chapter we present an overview of the CORAS risk assessment methodology and compare it with the TRESPASS process for risk assessment.

9.1. CORAS

CORAS is a security risk modelling language and the associated method for risk analysis (CORAS, 2016; Lund, Solhaug, & Stølen, 2011; Refsdal, Solhaug, & Stølen, 2015). For the deliverable to be self-contained, below we provide a summary of CORAS from D5.2.1 (The TRESPASS Project, D5.2.1, 2014).

CORAS, a method for conducting a risk analysis, is the result of a European funded project, lasting from January 2001 until September 2003 which had the goal to develop a tool-supported methodology for model-based risk analysis of security-critical systems.

CORAS is model-based and offers a customised language for threat and risk modelling and the corresponding guidelines on how to use the language. For modelling the target of the analysis, CORAS uses the Unified Modelling Language (UML).

The CORAS security risk analysis consists of eight different steps where the first four steps focus on context establishment and the last four steps are about risk identification, estimation, evaluation and possible risk treatments, see Figure 9.1.

In the following, the eight steps will be briefly described (descriptions based on (CORAS, 2016) and (Lund et al., 2011):

Step 1 - Preparations for the risk analysis: In order to prepare the risk analysis, this step focuses on defining the scope and estimating the size of the project.

Step 2 - Customer presentation of the target: This step consists of an introductory meeting with the customer. The main item on the agenda is a presentation of the responsible persons of the customer, revealing their general objectives and expectations and the exact scope of the risk analysis. This has the aim to give a common understanding of the scope and to identify what the targeted organisation is worried about.

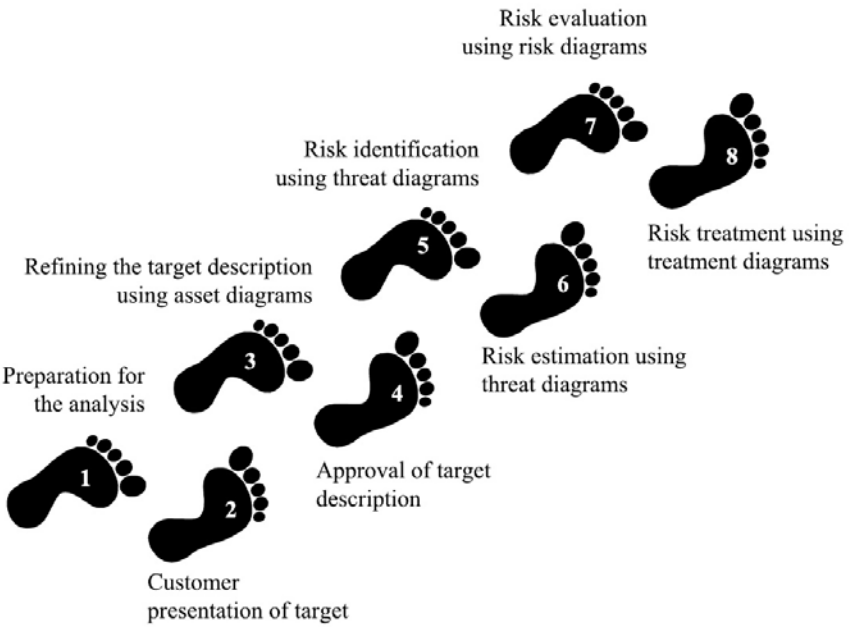


Figure 9.1.: The 8 steps of CORAS security analysis method (CORAS, 2016)

Step 3 - Refining the target description using asset diagrams: The goal of step 3 is to ensure a common understanding of the focus, the scope and the main assets. For this, the analysis team recapitulates the main results of the first meeting and the information from the readings of the company documents. Additionally the main assets to be protected are identified based on the interaction with the customer and a rough high-level analysis is conducted to identify major threat scenarios, vulnerabilities and enterprise risk levels.

Step 4 - Approval of the target description: Step 4 concludes the context establishment and includes, as a task, producing a detailed description of the scope of the risk analysis by using a formal or semi-formal notation such as the UML. The description should be approved by the customer before moving on to the next step. In addition, the definition of the risk evaluation criteria for each asset is also done during this step.

Step 5 - Risk identification using threat diagrams: Step 5 includes the identification of possible risks by organising a brainstorming meeting with participants who have different competences in order to identify as many risks as possible. Risk identification includes identification of threats, unwanted incidents, threat scenarios and vulnerabilities with references to the identified assets. The results will be documented with the help of CORAS threat diagrams, part of the CORAS language.

Step 6 - Risk estimation using threat diagrams: Step 6 takes the results from step 5 in order to define the level of the risks. Step 6 is, similarly to step 5, also conducted as a brainstorming with participants having different competences in order to estimate the likelihoods and consequences of unwanted incidents.

Step 7 - Risk evaluation using risk diagrams: Step 7 consists in evaluating if the identified risks are acceptable or not. The evaluation is done by using the risk evaluation criteria, defined during the context establishment and the results of risk estimations in step 6.

Step 8 - Risk treatment using treatment diagrams: The aim of step 8 is the identification of risk treatments for risks which are classified as not acceptable. Different risk treatments are chosen based on a cost-benefit analysis.

CORAS relies on its own modelling language which is an extension of UML. The methodology defines four kinds of diagrams (asset, threat, risk and treatment diagrams) as part of its "model-based" approach to support various visualisations in all steps of the process. These diagrams can be used in conjunction with the risk assessment to serve three purposes:

- Describing the target of assessment;
- As a communication medium that facilitates interaction between different groups of stakeholders;
- Documenting the results and underlying assumptions.

The method differentiates between direct and indirect assets (where assets are defined as entities that need to be protected). Furthermore, it classifies threats to these assets as:

- Human threat (accidental);
- Human threat (deliberate);
- Non-human threat.

The CORAS method is based on the ISO/IEC 17799 standard (now ISO/IEC 27002) and as such it is also compatible with ISO/IEC 13335 (now 27005), as well as the AS/NZS 4360 standard. Furthermore, CORAS provides a tool developed to be used together with the CORAS method (CORAS, 2016).

9.2. Comparison

We can notice that the sequence of CORAS steps presented above is actually a mixture of the risk assessment and analysis process (the CORAS *risk analysis process*) and the CORAS *service* that consists of specific interactions with a client. Therefore, we need to distinguish these two parts and to compare them separately with, respectively, the TRE$_S$PASS *risk analysis process* and the TRE$_S$PASS *service*.

Comparison of processes
Table 9.1 compares the risk assessment and analysis process steps in CORAS versus TRE$_S$PASS.

Table 9.1.: Steps in the CORAS and TRE$_S$PASS processes

CORAS steps	**TRE$_S$PASS steps**
(1) Informal creation of the target model and collection of relevant documentation	(1) Data gathering, knowledge base building, domain-specific tree bank population
(2) Design of the target model	(2) Model creation
(3) Asset identification and representation as asset diagrams	(3) Selection of relevant assets, scenarios and attacker profiles
(4a) Brainstorming on threats and vulnerabilities and preparing a high-level risk table (4b) Structured brainstorming to identify risks and create threat diagrams	(4a) Attack generation (for all scenarios identified in (3)) (4b) Attack tree extension and annotation
(5a) Structured brainstorming for risk estimation (estimation of consequences and likelihoods) (5b) Computing risk values	(5) Attack analysis
(6) Design of risk diagrams	(6) Visualisation of critical attacks and affected model elements
(7a) Brainstorming on risk treatments to reduce the risks to the acceptable levels (cost-benefit analysis) (7b) Introduction of treatments to threat diagrams and design of treatment overview diagrams	(7a) Human interpretation of the analysis results and countermeasure selection (cost-benefit analysis) (7b) Introduction of countermeasures to the model
	(8) (optional) Restart of the process from the model creation step (2)

Comparison of services
As an example of a TRE$_S$PASS service we consider the Audit use case scenario considered in (The TRE$_S$PASS Project, D6.2.2, 2015) and compare it to the CORAS risk analysis service. Table 9.2 summarizes both services.

9.3. Empirical Studies of Risk Assessment Methods

In this section we discuss how the TREsPASS process fares with respect to the criteria identified as important for security risk assessment methods in a series of empirical stud-

Table 9.2.: Phases in the CORAS and TRE$_S$PASS services

CORAS	**TRE$_S$PASS**
Meeting with the client to present the method, establish the goals and select the participants from the client and the analyst side (execution of step (1) of the CORAS process)	Meeting with the client to present the method, establish the goals and collect the relevant data (execution of step (1) of the TRE$_S$PASS process)
Independent execution of step (2) of the CORAS process by the analyst team	Independent execution of steps (1)-(7) of the TRE$_S$PASS process by the auditor
Meeting with the client to confirm and revise the target model and identify the assets (execution of step (3) of the CORAS process) (Remote) meeting with the client to confirm the target model and the asset diagrams, to identify data scales for consequence and likelihood and the risk evaluation criteria (step (4a) of the CORAS process) Independent execution of step (4b) of the CORAS process by the analysts Meeting with the client to estimate risks (perform step (5a) of the CORAS process) Independent execution of steps (5a) and (6) by the analysts Meeting with the client to present the evaluated risks and review the risk diagrams Meeting with the client to brainstorm on countermeasures (steps (7a) and (7b) of the CORAS process) and to present the final results	Meeting with the client to present the results and propose the relevant security controls identified
	(optional) Independent re-execution of the analysis by the consultant (optional) Meeting with the client to present the results

ies. The results presented of this section are published in (Gadyatskaya, Labunets, & Paci, 2016).

9.3.1. Criteria for security risk assessment methodologies

Labunets et al. (Labunets, Paci, Massacci, & Ruprai, 2014; Labunets, Massacci, Paci, et al., 2013; De Gramatica, Labunets, Massacci, Paci, & Tedeschi, 2015) have conducted a series of controlled experiments to investigate which are the main features of SRA methods that are behind the method's success. Success of a method is typically measured according to its *actual efficacy* in identifying threats and security controls and the *perceived efficacy* that participants have of the method, e.g. if they find the method easy to use or useful (Moody, 2003). To identify the features, Labunets et al. have applied qualitative analysis techniques from grounded theory to the interviews conducted with participants during the experiments. Four main features were identified that can determine the actual success of an SRA method.

Clear Process. Clear process means that the steps to identify assets, threats and security controls are well-defined and guidelines on how to apply the steps are provided to the analysts. If an SRA method has clear process, this positively affects the actual effectiveness of the method and the perception that the analysts have of the method. On the contrary, if the analysts do not know how a step of the process should be executed, the method will not be effective and will not be perceived as easy to use.

Visualization of risk models. Risk model visualization gives an overview of results of SRA, and thus may have a positive impact on an SRA method's success. However, if the visual notation does not scale for complex scenarios, it no longer provides a big picture of the risks threatening the target of analysis, and therefore it negatively affects the methods' effectiveness and perception.

Catalogues of threats and security controls. Catalogues can facilitate the identification of threats and controls especially for the analysts with limited security knowledge. As reported in (De Gramatica et al., 2015), domain experts without security expertise using domain-specific catalogues achieve better results than domain and security experts. Finding, sharing and validating threats and controls with catalogues is more efficient and effective, and, thus, the actual and perceived efficacy of an SRA method is higher.

Tool support. Tool can automatize the execution of an SRA process (e.g., computation of risk level) or can facilitate reporting of the results using an appropriate format (e.g. provide a set of tables that match method's steps). A well-designed tool can thus have a positive effect on method's success. In contrast, a primitive or buggy tool can only have a negative impact on the analysts perception of the method.

In addition to these main features, other important factors identified were:

Help in identifying threats and controls. Even if catalogues may not be included by default, the analysts appreciate if the methodology supports brainstorming and communication, and helps to elicit relevant threats and controls.

Change management and evolution support for SRA elements. The analysts appreciate if the method helps to ensure consistency across SRA elements (e.g. via traceability) when changes are introduced or the system evolves. This is especially important when dealing with large or evolving systems.

Scalability. For visual methods, such as CORAS and TREsPASS, scalability of diagrams becomes a challenge that, if not handled, can worsen method's effectiveness and perception.

9.3.2. CORAS evaluation findings

With respect to the criteria listed in the previous section, controlled experiments with CORAS have resulted in the following conclusions (Labunets et al., 2013, 2014).

Clear Process. The participants found the CORAS process to be clear and easy to use: "good methodology, not difficult to use. It is much clear to understand the security case there" in (Labunets et al., 2013). CORAS provides different types of diagrams that help practitioners to model the system and possible attack scenarios. However, some participants regarded that CORAS has redundant steps: "I think CORAS has some duplications" in (Labunets et al., 2014).

Visualization of risk models. CORAS enables a visual overview of the assets, possible sources of threats, threat scenarios and security controls, and helps the analysts to check that nothing has been overlooked: "diagrams are useful. You have an overview of the possible threat scenarios and you can find links among the scenarios" in (Labunets et al., 2013).

Catalogues of threats and security controls. CORAS does not include catalogues of threats and security controls. However, it can be used together with existing catalogues, e.g., BSI IT-Grundschutz or NIST-800-53.

Tool support. CORAS is supported by a diagram editor that helps to draw CORAS diagrams. However, the participants reported that the tool had low usability and was poorly developed. Thus, some of the participants acknowledged that they switched to an alternative solution for diagram drawing due to issues in using the tool (Labunets et al., 2014).

Help in identifying threats and controls. CORAS threat and treatment diagrams support the analysts in brainstorming threats and security controls.

Change management and evolution support for SRA elements. Once the analysis with CORAS is over, it can be hard for the analyst to update created diagrams. There is no traceability between diagrams in the tool. The participants in (Labunets et al., 2014) reported that in the CORAS tool "objects have no references between the diagrams. Changes on an object in a diagram are not reflected on the same object in other diagrams". Manual changes are time consuming and should be done carefully as it may affect many different diagrams.

Scalability. The participants of studies (Labunets et al., 2013, 2014) found the scalability issue to be relevant for CORAS: "these diagrams are getting soon very huge and very complex" in (Labunets et al., 2014).

9.3.3. Evaluation of TRESPASS

We now evaluate the TRESPASS methodology based on the criteria listed in Sec. 9.3.1.

Clear process. The studies (Labunets et al., 2013, 2014) mainly included novices in particular SRA methods (but not in information security), thus, the requirement of method and process clarity refers more to the question whether it is easy to master the method, than to whether a seasoned professional is able to achieve with it better results than with another method also familiar to him.

Visualization of risk models. The TRESPASS toolset supports hierarchical visualization of the system model and advanced visualization of attack scenarios (as paths on the system model, as well as attack trees) (Li, Barendse, Brodbeck, & Tanner, 2016; The TRESPASS Project, D4.3.2, 2016). The TRESPASS visualization capabilities have also been evaluated with security practitioners (Hall, Coles-Kemp, & Heath, 2016).

Catalogues of threats and security controls. In TRESPASS the role of catalogues is played by the knowledge base incorporating databases with relevant data, attacker profiles, and attack pattern library (tree banks). Furthermore, catalogues are used in TRESPASS to enable automated selection of countermeasures (Gadyatskaya, Harpes, Mauw, Muller, & Muller, 2016; The TRESPASS Project, D3.4.2, 2016). Thus, TRESPASS provides support for using existing knowledge in risk assessment.

Tool support. The TRESPASS methodology is supported by the TRESPASS toolset that provides support to the security analyst, as it automates some steps in risk assessment, as well as visualizes attack scenarios and the organization model. Yet, as studies (Labunets et al., 2013, 2014) reported, tools should not hinder the work of the analyst, and a buggy or unreliable tools may worsen the risk assessment results. Quality of the TRESPASS toolset is therefore of critical importance.

Help in identifying threats and controls. One of the main features of TRESPASS is the ability to automatically find attack scenarios based on the system model. This is of great value to the practitioners, as their workload is significantly reduced. Furthermore, TRESPASS partially supports automated selection of security controls and automated attack scenario identification (Gadyatskaya, Harpes, et al., 2016; The TRESPASS Project, D3.4.2, 2016).

Change management and evolution support for SRA elements. Risk assessment artifacts, such as threat diagrams in case of CORAS or system models in case of TRESPASS, are not stable but often need to be modified, e.g., when some earlier mistake or wrong assumption is identified. As both CORAS and TRESPASS are model-based, and they rely on model transformations as a part of their processes, change management is crucial. In this respect, TRESPASS includes elements of evolution support (The TRESPASS Project, D5.3.3, 2016). Since the attack generation part is automatic, if the organization model is changed or the considered attack scenario is revised, the discovered attack paths and analysis of those will be automatically re-computed. The identified critical attack paths will be mapped back to the organization model. Therefore, minor changes can be accommodated in the TRESPASS process seamlessly to the analyst. The evolution support is

further improved by maintaining more explicit traceability links among different underlying models and by improving the change management (The TRESPASS Project, D5.3.3, 2016).

Scalability. The TRESPASS tool adopts a scalable visualization approach, as it is able to zoom in and out the organization model. Moreover, the visualization of attacks is also scalable, as the analyst is presented with not a full generated attack tree, but with only the most important its parts (the zoom out feature for attack trees), or even only the critical attack paths, which are laid down in the organization model (The TRESPASS Project, D4.3.3, 2016). Furthermore, TRESPASS has investigated and proposed many methods for simplifying presentation of complex information (The TRESPASS Project, D4.3.2, 2016). Thus, the TRESPASS methodology, in principle, is able to deal with large use cases.

9.4. Conclusions and Lessons Learned

Based on our comparative study, we can conclude that CORAS and TRESPASS follow similar approaches to risk assessment. The main benefit of CORAS is its more advanced and structured service, i.e., the interaction with the client. Notice that the CORAS service is very well defined, including even the set of experts to be present at the meetings. In turn, the TRESPASS method is highly automated, therefore, it is less expensive. Moreover, it is less demanding in terms of expertise required from the client representatives attending the meetings.

In terms of the process-level differences, the TRESPASS method is less brainstorming-oriented than CORAS. This is an advantage (e.g., we can systematically find attack scenarios, rank them and propose mitigations), as well as a disadvantage (we can miss some attack scenarios that were not captured in the socio-technical model, but which could have been identified by the client). CORAS ensures mutual understanding for all stakeholders involved by the means of several diagram types used (target model, asset diagrams, threat diagrams, risk diagrams, treatment overview diagrams). The TRESPASS process instead uses only two main advanced visualization languages: the socio-technical model representing the organization, and the attack visualization in the Navigator (that includes advanced attack tree visualizations and visualization of attacks in the socio-technical model). Additionally, e^3fraud model also has a visualisation component.

Both CORAS and TRESPASS are visual methods and they have comparable processes, therefore it is possible to draw conclusions about these methods based on the criteria identified in empirical studies. Following the results of empirical studies with CORAS (Labunets et al., 2013, 2014), and the evaluation of TRESPASS reported above, we can summarize that the automation introduced in TRESPASS contributes to improvement of such features as *scalability, help in identifying threats and controls*, and *change management*. Furthermore, TRESPASS provides better support for *catalogues of threats and controls*, because it incorporates the knowledge base comprising threat-relevant information.

10. TRESPASS Process in Relation to FAIR

FAIR (Factor Analysis of Information Risk) is a Information Security Risk taxonomy at heart. It started from an industry-wide need for standardized terminology focused on *Risk* rather than *Security* (Jones, 2005)) and quickly evolved into a Technical Standard endorsed by The Open Group (Group, 2009) with an associated Risk Assessment Methodology (LLC, 2006) supported by an Excel-based tool (LLC, 2010).

The FAIR philosophy is based on the idea that "you can't effectively and consistently manage what you can't measure and you can't measure what you haven't defined" and therefore focuses on identifying, defining, classifying and relating the large variety of fundamental risk factors (see Figure 10.1). These factors, once quantified, can be used to compute overall risk levels. The FAIR Risk Assessment methodology provides guidelines on how to do these computations. The factors are measured on a 5-point ordinal scale and the computations are done with the help of look-up tables.

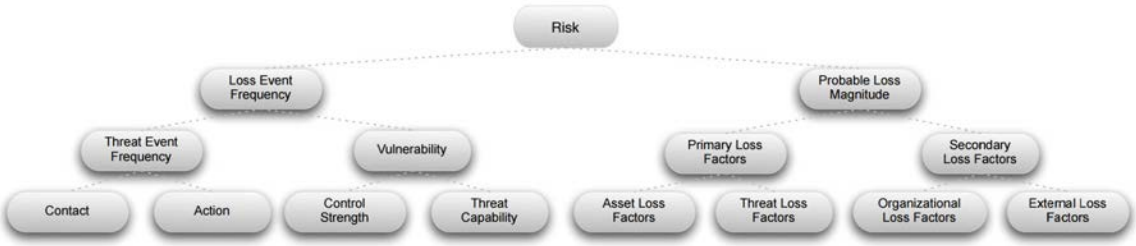


Figure 10.1.: FAIR's decomposition of Risk(LLC, 2006)

10.1. Risk factorization

FAIR's bottom-up computation of Risk is very similar to the TRESPASS approach. In fact, the TRESPASS glossary – a snapshot of which is presented in Section 1.2 of Deliverable 6.2.2.(The TRESPASS Project, D6.2.2, 2015) – is to a large extent inspired by FAIR's Risk Taxonomy.

However, while FAIR provides criteria for manually estimating risk factors on an ordinal scale, the TRESPASS toolkit provides tools that can automatically compute most of the same factors on numerical scales. Because the way TRESPASS computes risk factors is consistent with FAIR's taxonomy, one can use the FAIR criteria to cast TRESPASS output as input for a FAIR Risk Assessment. TRESPASS tools can therefore inform a

FAIR Risk assessment, while providing a higher level of precision and trace-ability of risk estimations.

10.2. Risk assessment process

The FAIR Risk Assessment methodology consists of four stages and a total of ten steps. By contrast, the TRE$_S$PASS approach consists also of four phases, but a much larger number of intermediary steps. This is mostly due to the different nature of the two approaches: FAIR attempts to guide a user through manually estimating individual risk factors, while TRE$_S$PASS aims to automate most of this estimation. FAIR estimations are based on intervals, criteria and rely on the user's intuitive knowledge while TRE$_S$PASS uses model, libraries and formal methods to lighten the load on the user.

Table 10.1 provides a mapping of the FAIR Risk Assessment stages and steps to the TRE$_S$PASS phases and activities. Please refer to the Integration Diagram in Deliverable 6.2.2 for a complete overview of the TRE$_S$PASS toolkit (The TRE$_S$PASS Project, D6.2.2, 2015).

Table 10.1.: A comparison of the TREsPASS and FAIR risk assessment processes

FAIR		**TREsPASS**	
stage	*step*	*activity*	*phase*
	N/A *(FAIR does not explicitly require data)*	Collect Technical data	Data collection phase
		Collect Physical Data	
		Collect Commercial Data	
Stage 1 – Identify scenario components	1. Identify the asset at risk	Collect Stakeholder Goals	
		Construct attacker goals	Model creation phase
	2. Identify the threat community under consideration	Attacker profile selection	
Stage 2 – Evaluate Loss Event Frequency (LEF)	3. Estimate the probable Threat Event Frequency (TEF)	**N/A** *(TREsPASS does not compute TEF)*	
	4. Estimate the Threat Capability (TCap)	Attacker profile creation	Model creation phase
	5. Estimate Control strength (CS)	Defender profile creation	
		TREsPASS model creation	
	6. Derive Vulnerability (Vuln)	TREsPASS analysis	Analysis phase
	7. Derive Loss Event Frequency (LEF)	**N/A** *(TREsPASS does not compute LEF)*	
Stage 3 – Evaluate Probable Loss Magnitude (PLM)	8. Estimate worst-case loss	**N/A** *(TREsPASS does not compute loss)*	
	9. Estimate probable loss	**N/A** *(TREsPASS does not compute loss)*	
Stage 4 – Derive and articulate Risk	10. Derive and articulate Risk	Attack tree analysis visualization	Visualization phase
	N/A *(FAIR does not expicitly prescribe any visualizations)*	Attack tree visualization	

Looking at Table 10.1, we can see some differences:

- FAIR skips the data collection steps. This is because TRE$_S$PASS requires a formal model of the Target of Assessment before any analyses can be carried out. For this reason, there is a bigger need for structured data when conducting a TRE$_S$PASS analysis.
- TRE$_S$PASS does not compute Threat Event Frequency. This is because TRE$_S$-PASS quantifies security risk as maximum utility for attackers. TRE$_S$PASS models do not supply attack frequencies, but only likelihood of success (given attack attempts) because we believe a priori frequencies are not the appropriate metric for adversarial risk (they depend on attacker strategy). The likelihood of success, together with the estimated costs and gains per attack vector provide a risk metric that can be used to rank attacks, without computing Threat Event Frequency.
- TRE$_S$PASS does not compute loss. This is because a TRE$_S$PASS analysis focuses on individual assets. In essence, TRE$_S$PASS focuses on detailing and improving the threat and vulnerability analyses, leaving impact assessment to be carried out manually. In order to obtain a complete risk picture, multiple assets would have to be analysed. The results can then be compared with regard to attacker utility: the asset with a high associated attacker utility may require more protection. Furthermore, TRE$_S$PASS would then support the calculation of which countermeasures provide reduced attacker utility over all possible targets. However, these results can be enhanced with the help of FAIR's (or other methodologies') impact estimation tables in order to gain an even more complete risk picture.
- FAIR does not explicitly prescribe any visualisations. One of the strong points of TRE$_S$PASS analyses, and an advantage of them relying on hard data rather than just estimates, is the large variety of visualisations they can support. TRE$_S$PASS tools can visualize attack vectors directly on the navigator map, can compare different attacks and vulnerabilities on a variety of metrics, and are able to run sensitivity analyses at the push of a button.

11. TRESPASS Process in Relation to TRICK Service

11.1. TRICK Service

11.1.1. Context

TRICK Service (Tool for Risk management of an ISMS based on a Central Knowledge base) is a risk assessment and management tool developed by itrust consulting for identification, analysis and estimation of assets, threats, vulnerabilities, risk scenarios and security measures. TRICK Service enables the determination of a list of security measures to implement in order to reduce the impact or the occurrence likelihood of possible risk scenarios.

11.1.2. History

Information Security Management Systems (ISMS) are becoming widely used, since the requirements of an ISMS have been defined by ISO 27001 in 2005, providing background for security certification similar to ISO 9001's quality certification.

itrust has chosen TRICK as acronym for "Tool for Risk management of an ISMS based on a Central Knowledge base" in the context of an FP7 project proposal. This proposal intended to develop a software tool and an inherent methodology to fix risk management problems encountered by participating SMEs. This tool is intended to be used across many countries, and enables consultants to share experience and figures automatically. It focuses on fast but quantitative risk evaluation. It models security measures with risk reduction properties that can easily be tailored to a specific organisation. It integrates know-how from multiple standards such as ISO 27002 and risk assessment methods like the French EBIOS and the German IT Grundschutz, with experiences of consultants in one central knowledge base, based on a high-level risk management language defined herein. It develops algorithms to assess risks and to derive residual risk after security implementation, and to compute their Return On Security Investment (ROSI). It develops a concept for easy adaptability to new standards and lessons learned from exploitation.

In 2007, itrust consulting decided to develop a prototype called TRICK light in the context of the internship of a master student. This first version was limited to a predefined asset and scenario list, grouped by types.

In the context of the CELTIC project BUGYO, a new version was developed enabling risk assessment for multiple assets, use of tailored risk scenarios, generation of risk treatment plans and statement of applicability for ISO 27001 certification. It also enables easy usability as it comes in the form of risk treatment plans and summary charts.

Due to several reasons such as performance and stability, the Excel version was replaced by a web application called from then on TRICK Service. TRICK Service includes all functionalities of the Excel version and was extended by a multitude of functionalities in the context of further research projects such as TRE$_S$PASS and SGL Cockpit. These functionalities include support for CSSF 12/544, required for Financial Service Providers (FSP), maturity assessments, sectoral catalogues of security controls and an improved interface.

11.1.3. Concept

Figure 11.1 below provides an overview of the different steps implemented in TRICK Service.

The different concept elements will be described in the following sections.

11.1.3.1. Risk analysis

Defining the context. We first define the context of the risk analysis by collecting information about the type and business processes of the studied organisation. This information will be used in the following steps by the analyst in order to evaluate what are the most important assets regarding the sector of the organisation. Defining the context consists of filling in a table containing all topics to be addressed in the context establishment according to ISO/IEC 27005:2011 (ISO/IEC, 2011).

After the context definition, a brainstorming session is triggered which produces a collection of all assets and risk scenarios in the organisation, serving as input for the next sub-step, namely asset identification.

Identifying the assets. This step consists of creating an inventory of the organisation's assets considered as important for the organisation's business. The assets are identified by name and grouped by type; the idea is to group assets together if they have the same risk profile. Other information that should be defined is the values of the assets and a comment in order to describe the asset or justify its value.

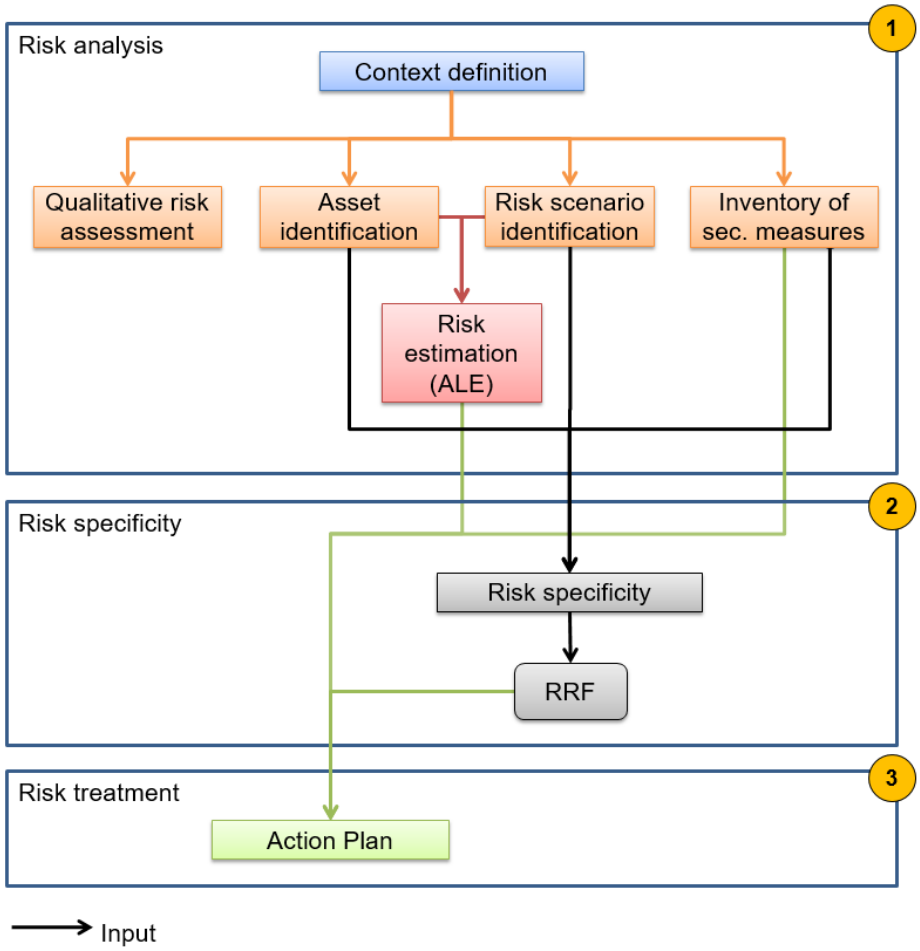


Figure 11.1.: TRICK Service concept.

Identifying different risk scenarios. As for asset identification, scenario identification consists of retrieving a majority of information concerning possible risk scenarios that could occur on the assets of the organisation. This identification of scenarios must take into account the context defined previously (e.g. an organisation located on a seismic zone should consider the destruction of its premises as a potential scenario). Eight generic risk scenarios presented in Table 11.1 are provided by default.

Inventory of security measures. This step consists of defining the current implementation rate and the cost for each security measure using several norms, including ISO/IEC 27002 (ISO/IEC, 2013). The way standards are modelled in TRICK Service allows the user to specify custom standards as well, which considerably increases the flexibility of the tool. The implementation rate allows the user to identify the security measures that are fully implemented so that only the remaining ones need to be considered for the risk treatment plan. It is also possible to exclude some measures which would not be relevant regarding the organisation context or to consider some measures as mandatory in order

Acronym	Name	Scenario type	Description
A_1	Partial loss or temporary	Availability	A part of the asset is lost or the asset is temporarily non-operational.
A_all	Complete loss, including backup	Availability	Loss of the whole asset, including backup.
C1	Partial theft coming from external	Confidentiality	An essential part of an asset was stolen without complicity of an internal person.
C2	Deliberate disclosure	Confidentiality	Internal staff copies the entire asset to disclose it.
C3	Accidental disclosure	Confidentiality	Following a false handling, an important part becomes accessible to people that are not authorised.
I1	External manipulation	Integrity	An external person succeeds penetrating and handling an asset.
I2	Fraudulent manipulation coming from internal	Integrity	An internal person handles an asset to create an illicit advantage.
I3	Accidental manipulation	Integrity	A technical or organisational error causes a corruption of an asset.

Table 11.1.: Default generic risk scenarios.

to force them to appear at the beginning of the action plan and in order to manage the case where there are dependencies between measures.

Risk estimation. This step is one of the most important of the TRICK Service process. The analyst discusses, analyses and estimates the annual loss expectancy (ALE) of each couple asset/scenario. The ALE is computed by multiplying the impact (in euros) that a scenario could have if occurring on an asset, with the potentiality (expressed as expected frequency per year) that a scenario could occur on an asset. The impact estimation criteria and likelihood estimation criteria are respectively listed in tables 11.2 and 11.3 below, and their values are customisable.

For companies with the Professional of the Financial Sector (PFS) status, some additional CSSF impact estimation criteria must be used.

Impact level					
Level	Acronym	Qualification	Value (k€)	Range min	Range max
0	i_0	Insignificant	1	0	1
1	i_1		2	1	2
2	i_2	Minor	3	2	4
3	i_3		5	4	7
4	i_4	Serious	10	7	13
5	i_5		17	13	23
6	i_6	Very serious	30	23	41
7	i_7		55	41	74
8	i_8	Extremely serious	100	73	132
9	i_9		173	132	228
10	i_{10}	Vital	300	228	$+\infty$

Table 11.2.: Impact estimation criteria.

Probability of threat occurrence					
Level	Acronym	Qualification	Value	Range min	Range max
0	p_0	Never (or less than every 30 years)	0.02/y	0.00	0.04
1	p_1	$A\ priori$ not occurring, very improbable	0.04/y	0.04	0.08
2	p_2	Isolated, rare, once every 10 years	0.10/y	0.08	0.14
3	p_3		0.18/y	0.14	0.25
4	p_4	Repetitive, possible, once every 3 years	0.33/y	0.25	0.44
5	p_5		0.58/y	0.44	0.76
6	p_6	Recurring, probable, once every year	1.00/y	0.76	1.41
7	p_7		2.00/y	1.41	2.83
8	p_8	Common, very probable, once every quarter	4.00/y	2.83	5.26
9	p_9		6.93/y	5.26	9.12
10	p_{10}	Constant, certain, once a month	12.00/y	9.12	$+\infty$

Table 11.3.: Likelihood estimation criteria.

11.1.3.2. Risk specificity

Risk specificity criteria. The objective of this step is to quantify the risk specificity for the generation of the RRF. We apply risk specificity to three elements:

- For each **risk scenario**, we determine if it specifically relates to confidentiality, integrity or availability, if it is of intentional, accidental or environmental cause, etc.;
- For each **security measure**, we qualify their influences on every security criteria;
- For **assets**, we directly define the influence of each security measure on each asset.

The risk specificity step has partially already been performed by the analyst and specificity values are freely available for the security measures of ISO/IEC 27001 and ISO/IEC 27002, and the asset types. The only elements which require a definition during a risk analysis are the risk specificity of risk scenarios added by the user and the risk specificity of customised security measures not covered by the standard libraries, such as ISO/IEC 27001 and ISO/IEC 27002.

Tables 11.4, 11.5, 11.6 and 11.7 describe the different risk specificities in detail. Each of the tables indicates to what element (measure, scenario, or asset) the risk specificity applies.

Group	Criterion	Measure	Scenario	Asset	Formula	Description
Asset type/ measure	Asset type/ measure	X		X	A_t/M	This criterion defines the level of influence of a security measure on a type of asset. Accepted values: decimal between 0 and 1.
Strength	General strength of the measure	X			SG_m	Defines the generic influence to the participation in the improvement of the security if a measure is implemented. Accepted values: integer between 0 and 10.
	Sectorial strength of the measure	X			SS_m	Allows moderating generic influence of a measure, considering a particular sector (SME, bank, etc.). Accepted values: integer between 0 and 4.

Table 11.4.: Risk specificity "Asset type/measure" and "Strength".

Group	Criterion	Measure	Scenario	Asset	Formula	Description
Category	Confidentiality	X	X		C_s	Applied to a scenario, it defines its level of impact on the confidentiality of assets. Accepted values: 0 or 1.
					C_m	Applied to a security measure, it defines its level of influence for the protection of the confidentiality of assets. Accepted values: integer between 0 and 4.
	Integrity	X	X		I_s	Applied to a scenario, it defines its level of impact on the integrity of assets. Accepted values: 0 or 1.
					I_m	Applied to a security measure, it defines its level of influence for the protection of the integrity of assets. Accepted values: integer between 0 and 4.
	Availability	X	X		A_s	Applied to a scenario, it defines its level of impact on the availability of the assets. Accepted values: 0 or 1.
					A_m	Applied to a security measure, it defines its level of influence for the protection of the availability of assets. Accepted values: integer between 0 and 4.
	CSSF categories	X	X		$D1_s$ to $I10_s$	When a scenario belongs to a CSSF type, its corresponding CSSF category equals to one and all other CSSF categories equals to zero. When a scenario does not belong to a CSSF type, all CSSF category equals to zero.
					$D1_m$ to $I10_m$	Applied to a security measure, it defines whether it has an influence on each CSSF category. Accepted values: integer between 0 and 4.

Table 11.5.: Risk specificity "Category".

CSSF is the financial sector regulator in Luxembourg. Considerations of specific risk categories imposed by the regulator have been added to TRICK Service thanks to the TRE$_S$PASS project.

Group	Criterion	Measure	Scenario	Asset	Formula	Description
Type	Preventive	X	X		$Prev_s$	Characterises if the impact of a scenario is affected by preventive security measures. Accepted values: 0 or 1.
					$Prev_m$	Characterises the preventive property of a security measure. Accepted values: integer between 0 and 4.
	Detective	X	X		Det_s	Characterises if the impact of a scenario is affected by detective security measures. Accepted values: 0 or 1.
					Det_m	Characterises the detective property of a security measure. Accepted values: integer between 0 and 4.
	Limitative	X	X		Lim_s	Characterises if the impact of a scenario is affected by limitative security measures. Accepted values: 0 or 1.
					Lim_m	Characterises the limitative property of a security measure. Accepted values: integer between 0 and 4.
	Corrective	X	X		Cor_s	Characterises if the impact of a scenario is affected by corrective security measures. Accepted values: 0 or 1. Note for the scenario type properties: the total sum of the four criteria must equal to one: $Prev_s + Det_s + Lim_s + Cor_s = 1$
					Cor_m	Characterises the corrective property of a security measure. Accepted values: integer between 0 and 4.

Table 11.6.: Risk specificity "Type".

Group	Criterion	Measure	Scenario	Asset	Formula	Description
Source	Intentional	X	X		Int_s	Applied to a scenario, this criterion defines if the scenario source is intentional. Accepted values: 0 or 1.
					Int_m	Applied to a security measure, this criterion defines its influence on the scenarios of intentional type. Accepted values: integer between 0 and 4.
	Accidental	X	X		Acc_s	Applied to a scenario, this criterion defines if the scenario source is accidental. Accepted values: 0 or 1.
					Acc_m	Applied to a security measure, this criterion defines its influence on the scenarios of accidental type. Accepted values: integer between 0 and 4.
	Environmental	X	X		Env_s	Applied to a scenario, this criterion defines if the scenario source is environmental. Accepted values: 0 or 1.
					Env_m	Applied to a security measure, this criterion defines its influence on the scenarios of environmental type. Accepted values: integer between 0 and 4.
	Internal threat	X	X		$IntT_s$	Applied to a scenario, this criterion defines if the scenario source is internal to the organisation. Accepted values: 0 or 1.
					$IntT_m$	Applied to a security measure, this criterion defines its influence on the scenarios of internal type. Accepted values: integer between 0 and 4.
	External threat	X	X		$ExtT_s$	Applied to a scenario, this criterion defines if the scenario source is external to the organisation.
					$ExtT_m$	Applied to a security measure, this criterion defines its influence on the scenarios of external type. Accepted values: integer between 0 and 4.
Max RRF	-				-	Max RRF value (between 0 and 1) is modifiable by user in order to globally adapt the influence of the RRF.

Table 11.7.: Risk specificity "Source".

Risk Reduction Factors. Risk specificity criteria used to compute the RRFs are sorted by group in Figure 11.2.

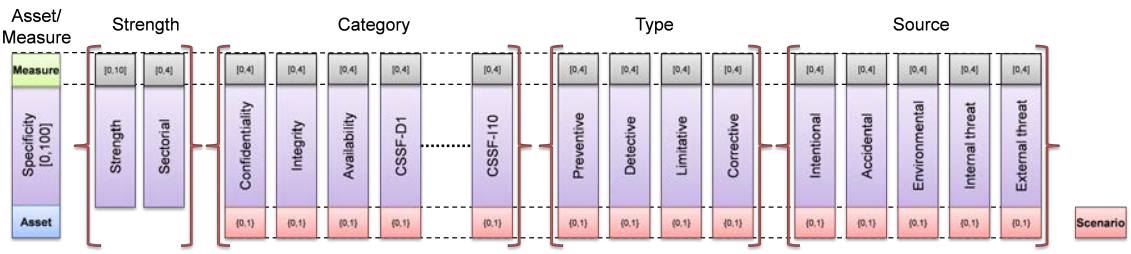


Figure 11.2.: Risk specificity criteria used to compute RRFs.

All previously mentioned security criteria are now used to compute the Risk Reduction Factor (RRF) associated to each asset-scenario-measure triple. Concretely, by associating these criteria together using a weighted computing, we determine a global coefficient of the influence of the security measures on the ALE generated by the occurrence of a scenario on an asset. The RRF thus represents the ALE reduction generated by complete implementation of the security measure, i.e. for a given security control in relation to a given scenario acting on an asset, its RRF is a value between 0 and 1, where RRF=0 means that the countermeasure is useless, and RRF=1 signifies perfect protection.

11.1.3.3. Risk treatment

Difference of ALE for the implementation of a measure on an asset for a given scenario. The $\Delta ALE_{s,m,a}$ is the ALE reduction generated by the complete implementation of a measure on the ALE created by the occurrence of a scenario on an asset. Consider $ALE_{s,a}^{start}$ the ALE generated by the occurrence of a scenario s on an asset a, consider IR_m the implementation rate of a security measure m, and $RRF_{s,m,a}$ the RRF associated to the triple (s, m, a), the formula used to compute the ALE reduction generated by the measure m on the ALE created by the scenario s on the asset a is the following:

$$\Delta ALE_{s,m,a} = ALE_{s,a}^{start} \times RRF_{s,m,a} \times \frac{1 - IR_m}{1 - RRF_{s,m,a} \times IR_m}$$

We deduce the general ΔALE_m generated by the full implementation of a measure:

$$\Delta ALE_m = \sum_{s,a} \Delta ALE_{s,m,a}$$

In this document, we can skip the implementation rates of security measures, assuming that our model proposes to fully implement a countermeasure that was not implemented before.

ROSI. The Return On Security Investment (ROSI) is based on the ROI concept, which consists of investing a sum and gaining at least the equivalent, the ideal being to pass the invest sum by a maximal margin.

$$ROI = Return - Investment$$

Example 1: For €4,000 invested and €10,000 of return, the ROSI would be of €10,000 – €4,000 = €6,000.

The reasoning of the ROSI considers the investment made when implementing the security measure: by analogy with the ROI, the cost of the implementation of a measure corresponds to the invested sum and the ΔALE_M corresponds to the gains. Thus, we have the following formula:

$$ROSI_M = \Delta ALE_M - cost_M$$

Example 2: Considering the scenario "deletion of data" which impacts the "know-how" of an organisation, it results in an ALE which can be estimated at €100,000. The whole implementation of a solution of "data backup" would enable a decrease of the ALE by €75,000 (ΔALE_M). Knowing that the cost of the implementation of this measure of backup is €5,000 ($cost_M$), we have a $ROSI_M$ of €70,000 ($\Delta ALE_M - cost_M$).

Choice of measures. The selection of the security measures to implement is made by comparing the $ROSI_M$ of each available security measure: concretely, we compute the $ROSI_M$ of each measure and we choose the one with the maximum ROSI. In the case where the maximum ROSI is superior to 0, we add it to the list of measures to implement and we continue with the same comparison with the other measures. If the maximum ROSI is negative and no mandatory measures remain, we stop the computation because there is no relevance to select a measure whose ΔALE_M is lower than the cost of the implementation. In other words, any measures with negative ROSI are discarded from the last implementation phase.

11.2. Extension to TRICK Service with attack-defence trees

The purpose of attack-defence trees (ADTrees) is to model attack-defence scenarios, which can be seen as a game between two players, the proponent and the opponent. When the root of the tree is an attack node, the proponent is an attacker and the opponent is a defender, and the opposite when the root is a defence node. The children of the root in an AD tree are refinements of the global goal of the proponent.

When drawing ADTrees, attack nodes are depicted by red circles (◯) and defence nodes by green rectangles (□). Refinement relations are indicated by solid edges between nodes and countermeasures are indicated by dotted edges. A conjunctive refinement of a node is depicted by an arc over all edges connecting the node and its children of equal type. To construct ADTrees, we first start drawing the root of the tree, which represents the main goal of the attack-defence scenario. This goal is then refined into subgoals represented by children of the root which in turn may get further refined. Countermeasures can be included by inserting one child of the opposite type, which may also be refined into specific goals and so on.

itrust consulting proposed the use of an ATree of a risk scenario to fine-tune risk assessment of an organisation with TRICK Service. The analyst who is using TRICK Service can now express threat scenarios as attack trees, and perform the subsequent risk treatment steps using these trees.

11.3. Implementation

For a concrete implementation, ADTop (Attack-Defence Tree optimizer) has been developed as a transition software tool and integrated with the ADTool format, in order to bridge the gap between the theoretical model of attack-defence trees and concrete risk analysis coming from TRICK Service. The full approach is described in more details in D3.4.2 (The TRESPASS Project, D3.4.2, 2016).

11.3.1. High-level process

ADTop receives an ATree and an extract of a risk analysis. It generates an association matrix, which helps to perform calculations for the optimal selection of preventive security controls, and produces an optimal ADTree.

11.3.2. Software architecture and workflow

Figure 11.3 shows the different software tools and their interactions.

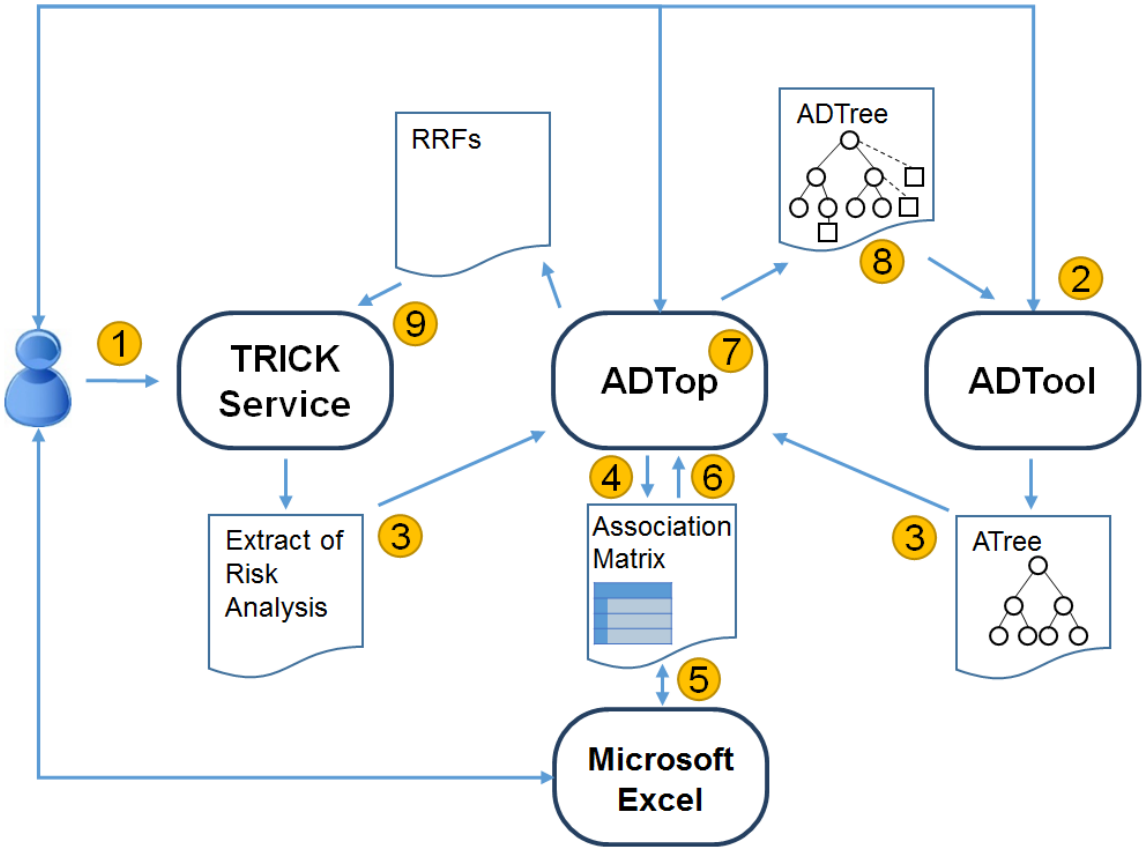


Figure 11.3.: Software architecture and workflow.

1. The analyst initiates the risk assessment process. With help of standards or lists of measures, he/she builds a risk analysis in TRICK Service for one specific risk scenario applied to one given asset.
2. With help of ADTool, the analyst builds an initial ATree, which includes the relevant threats according to the previous risk scenario and context of the risk analysis.
3. The ATree and information extracted from the risk analysis (see the description of files in section 11.3.3.3) are imported into ADTop.
4. ADTop processes the information collected in the previous step and builds an association matrix of attacks and countermeasures.
5. The analyst manually enters relevant effectiveness values of countermeasures in the association matrix, everytime a countermeasure is applicable to a given attack.
6. The association matrix is re-imported and all the countermeasures selected are added into the initial ATree (see section 11.3.3.5 for more details).
7. ADTop performs calculations based on the ROSI function. At the moment, this step consists of evaluating all combinations of attacks and countermeasures, which is equivalent to a brute-force algorithm (see section 11.3.3.6).
8. ADTop generates optimal ADTrees, which can be visualised in ADTool.
9. ADTop exports a file containing the generated RRFs for one specific risk scenario applied to one given asset in JSON format, that can further be compared with those of TRICK Service.

11.3.3. Detailed process specifications

11.3.3.1. Interactions

The analyst uses the following software tools and applications: ADTool, ADTop, TRICK Service, and Microsoft Excel. Microsoft Excel enables the modification of the association matrix. ADTool and ADTop communicate between each other by means of files of the same format. ADTop and TRICK Service also communicate the same way or by means of an Application Programming Interface (API).

11.3.3.2. Input/Output

- ADTool imports and exports files containing ATrees or ADTrees in XML format;
- ADTop imports an extract of the risk analysis coming from TRICK Service in JSON format. Alternatively, ADTop can import the extract of risk analysis from TRICK Service, thanks to an API;
- ADTop exports a file containing the optimal ADTree in XML format. ADTop is also able to export different files containing optimal ADTrees selected by the analyst;

- ADTop exports and imports a file containing an association matrix of attacks and countermeasures in XLS format;
- TRICK Service exports a file containing information from its risk analysis in JSON format.

11.3.3.3. File content

The file containing the initial ATree has the following relevant information:

- ATree structure;
- Name of the attacks;
- Success probability of the leaf nodes (atomic attacks).

The file containing the optimal ADTree has the following information:

- ADTree structure;
- Name of the attacks;
- Name of the countermeasures;
- Success probability of the attacks;
- Effectiveness value of the countermeasures;
- Security implementation cost of the countermeasures.

The extract of the risk analysis contains the following relevant information:

- Annual Loss Expectancy (ALE), which corresponds to the global impact of a risk scenario, e.g. two millions euros for data leak;
- Name of the scenario;
- Name of the security controls;
- Implementation rate of the security controls;
- Security implementation cost of the security controls.

The association matrix contains the following information:

- Name of the countermeasures;
- Name of the attacks to which the countermeasures are applicable;
- Effectiveness value of countermeasures for the attacks they refer to.

11.3.3.4. Justification of file formats

JavaScript Object Notation (JSON) is a recent lightweight text-based open standard format designed for human-readable data interchange; the most common data format used for asynchronous browser/server communication. It is derived from the JavaScript programming language for representing simple objects. It has been chosen as an alternative of XML because it is efficient, language-independent, with parsers available for most programming languages.

Excel Binary File Format (XLS) has been chosen because it is simple and widely used to manage data within organisations. It helps the analyst to manually modify the different effectiveness values of the countermeasures in the association matrix.

Extensible Markup Language (XML) has been chosen because it is both a human-readable and machine-readable format, which emphasises simplicity and usability across the Internet, and because it is the format used for the files managed by ADTool.

11.3.3.5. Adding countermeasures to the ATree

1. Starting from the root of the imported initial ATree, ADTop takes the attack name and looks for the same attack name in the association matrix.
2. With the corresponding attack found in the the association matrix (reference column), ADTop browses all the corresponding countermeasures (rows) and adds them to the current node. Adding countermeasures happens everytime a non-blank effectiveness value is found. If several countermeasures apply to the same attack, ADTop builds a set of countermeasures instead of only adding one countermeasure to the current node, which follows the process based on our assumptions on countermeasure selection and combinations of defence nodes described in our approach.
3. ADTop continues to browse the ATree until all leaf nodes (atomic attacks) are reached and repeats the process described in the previous step. If a countermeasure has already been selected, ADTop doesn't add it to the current node.

11.3.3.6. Branching algorithm

For the initial implementation with ADTop, the ROSI-based optimisation problem is solved thanks to a branching algorithm. Branching alone is equivalent to brute-forcing all the candidate solutions. The ROSI optimisation consist in minimising the residual success probability of the attack scenario as well as minimising the sum of the security implementation costs of the selected countermeasures. To be able to find optimal countermeasures for large, complex scenarios, and to improve the performance of brute-force search, we need to replace the exhaustive search with a more efficient algorithm or a heuristic described in the next section.

11.3.4. Options for improvement

A Branch and Bound algorithm (or B&B algorithm) keeps track of bounds on the minimum that it is trying to find, and uses these bounds to prune the search space, eliminating candidate solutions which will not contain an optimal solution. This algorithm will limit the number of candidate solutions to be explored. Bounding is only possible if we know in advance that a countermeasure will not be selected. Our case is more complicated, thus the B&B algorithm used in (Roy, Kim, & Trivedi, 2012) is not applicable.

However, in cooperation with the research project IDS4ICS by itrust consulting, an optimisation algorithm has been found (Muller, Harpes, & Muller, 2016). An adjustment of the recursive branching algorithm is possible by skipping all sets of combinations that are known not to contain any solutions. The algorithm will never skip a valid combination. The idea is to skip a recursion step when it does not provide a viable combination of selected countermeasures. Whenever a defence is added to an attack-defence tree, the success probability of any attack node decreases. The same is true for the global success probability of the tree. When the algorithm enters a recursion step, all non-processed countermeasures are set to 'unselected'. Thus, all deeper recursion steps will end up with a lower or equal overall success probability for the attack-defence tree. As a consequence, once the probability is no longer sufficiently reduced to cover the costs, i.e. once a defence is no longer profitable, it will not be profitable for all later combinations either, which means that all subsequent combinations, that are *a priori* known to be invalid, can be skipped.

Algorithm 1 Branch and bound algorithm BnBa

Input: Attack–defence tree T with attack nodes A
Input: Set of defences D
Input: Effectiveness values $e : A \times D \to [0, 1]$
Input: Set of already processed defences $D_p \subseteq D$
Input: Partial selection strategy $x : D_p \to \{0, 1\}$
Output: Selection strategy x_{opt} that maximises ROSI$(\cdot)$

if there is $\delta \in D_p$ that is no longer profitable (cf. Algorithm 2) **then**
  abort current recursion step
end if
if $D_p = D$ **then**
  $v \leftarrow$ **ROSI**(x)
  if v is largest ROSI seen so far **then**
    $x_{\text{opt}} \leftarrow x$
  end if
else
  $\delta \leftarrow$ any defence *not* in D_p
  $D_p \leftarrow D_p \cup \{\delta\}$
  ' Try selecting the defence
  $x(\delta) \leftarrow 1$
  BnBa(T, D, e, D_p, x)
  ' Try not selecting the defence
  $x(\delta) \leftarrow 0$
  BnBa(T, D, e, D_p, x)
  ' Remove δ again; this allows the re-use of D_p among all recursive calls
  $D_p \leftarrow D_p \setminus \{\delta\}$
end if

Algorithm 2 Determine if a defence is profitable

Input: Defence δ
Input: Cost $c(\delta)$ of defence δ
Input: Impact $\mathcal{I}$ of risk scenario
Input: Partial selection strategy $x : D_p \rightarrow \{0, 1\}$
Output: **true** if δ is profitable, **false** otherwise

if $x(\delta) = 0$ **then**
    return true
else
    ' Extend x to all of D
    $x(\delta') \leftarrow 0$ for all $\delta' \in D \setminus D_p$
    $x(\delta) \leftarrow 0$
    $v_0 \leftarrow \text{ROSI}(x)$
    $x(\delta) \leftarrow 1$
    $v_1 \leftarrow \text{ROSI}(x)$
    ' δ is profitable iff the residual risk is lower when δ is implemented
    if $v_1 \cdot \mathcal{I} + c(\delta) < v_0 \cdot \mathcal{I}$ **then**
        return true
    else
        return false
    end if
end if

We may also consider different constraints, e.g. the maximum budget for the implementation of preventive security controls and the acceptable residual success probability for the risk scenario set by the analyst, the number of iterations that can be determined by the algorithm itself to comply with an acceptable error rate under five percent, etc., in order to reduce the explored space.

11.4. Conclusions

itrust consulting has investigated the problem of automated countermeasure selection in risk treatment. The approach presented in D3.4.2, together with the new algorithm proposal and the add of constraints for the security implementation budget and acceptable residual success probability, brings together the well-known attack-defence tree methodology and a practical security risk management tool, TRICK Service, in order to select the optimal countermeasures to be implemented in an organisation. It also helps boosting the performance of the ADTop tool, and determining a set of optimal ISO/IEC 27002 security controls that have the largest added-value for the realistic ÉpStan Trusted Third Party (ÉpStan TTP) case study of the University of Luxembourg, which deals with the implementation of a pseudonymisation service, and described in D7.4.2 (The TRESPASS Project, D7.4.2, 2016).

12. The TRESPASS Service Model

This chapter provides an overview of a possible model for offering TRESPASS risk analysis as a service. The model is based on the navigation metaphor as explained in Pieters, Barendse, Ford, Heath, and Probst (2016).

12.1. The navigation metaphor

From the start of TRESPASS, the concept of the "attack navigator" has been central to tool development. However, during the course of the project, the process around the tools has also been devised increasingly around what we call "the navigation metaphor" in cyber security, in particular in security economics. The key feature of the metaphor is that it enables the expression of stages in the security risk assessment process in terms of maps and routes. Within this process, TRESPASS languages, visualisations, and analyses can serve to support the understanding of the security of the target of analysis in terms of navigation by attackers, and "road blocks" by defenders.

We believe that this framing of the risk assessment process provides an excellent basis for service models, in which the TRESPASS risk assessment process, in different forms, can be offered as a service to clients, primarily medium-sized to large organisations. The benefits are not only the outcome of the risk analysis, but also additional awareness and understanding of the risk picture in the organisation by participation of client personnel in the various stages of development.

Based on the different stages in developing and using a navigation system, the service offering can be explained in phases, including activities of the client, of the consultant, joint meetings, and results.

12.2. Service model and interaction with the client

In this section, we outline the stages of the navigation metaphor as described in (Pieters et al., 2016). We focus specifically on the service model around these stages, which is not included in the scientific paper.

12.2.1. Satellite view

The goal of the satellite view stage is developing a common understanding of the target of analysis, its socio-technical architecture, and relevant security aspects.

Possible services that could be part of this stage are:

- LEGO modelling exercise
- modelling exercise with cards
- security argumentation game

An overview of these best practices is described in The TRESPASS Project, D5.3.2 (2015), More details about LEGO can be found in the WP4 deliverables.

In these activities, TRESPASS consultants typically facilitate the building of a shared understanding of the system at hand by client representatives, who engage in the hands-on development of representations.

The result of this stage is an informal or semi-formal representation of the security landscape of the target of analysis. This will be documented in a report plus pictures of the designed representation (e.g. LEGO or cards model). The result should also include an informal description of the attacker personas, as well as their goals: who does the client see as potential threats to their systems (insiders as well as outsiders), and what could they be after?

These activities are generally labour-intensive, and would require at least a full day of interaction with the client.

12.2.2. Map

The map stage generally requires less interaction with the client. In this stage, the TRESPASS consultants formalise the satellite view into a map, similar to the rendering of geographical maps from satellite images. The TRESPASS modelling language (The TRESPASS Project, D1.2.2, 2015; The TRESPASS Project, D1.3.4, 2016) defines the available map elements and their connections, such as actors, locations, etc.

In addition, this stage defines the formal properties of the attacker profiles based on the attacker personas (The TRESPASS Project, D5.3.2, 2015). These would correspond to properties of the car in geographical navigation systems (e.g. whether the car has 4WD).

Finally, the consultants prepare the relevant libraries, such as the Attack Pattern Library (The TRESPASS Project, D5.3.2, 2015) and the Knowledge Base (The TRESPASS Project, D2.4.1, 2016), for the specific case. Additional case-specific information may need to be obtained from the client.

If there is sufficient modelling expertise on the client side, the result of this stage (navigator map) can be discussed with the client in a shorter session. The main goal is to gather

feedback from the client in terms of correctness and completeness of the map. Such a session would require 2 hours up to half a day.

12.2.3. Routes

After the map is ready, the TRESPASS analysis tools are invoked to calculate likely routes for attacker personas to reach their goals.

The most likely routes should be presented to the client in an evaluation session, in which the client can evaluate whether the outcomes are perceived as realistic. If not, then possible problems in the routes are identified, and these are mapped back to the corresponding components of the map, which may require changes. Depending on the number of scenarios and the amount of feedback, such a session would typically require around half a day.

12.2.4. Optimisation

Together with the client, possible defensive measures are discussed, based on the weak links in the scenarios. A set of defensive measures is agreed upon. This session would be around 2 hours.

The TRESPASS consultants then translate these measures into changes on the map, and re-evaluate the routes / scenarios in order to assess the effect of the changes on security.

A final meeting is set up with the client in which recommendations are given on suitable measures. When feasible, further real-time iterations with adaptations in countermeasures can be performed in this meeting to incorporate wishes of the client.

12.3. Conclusion

In this chapter, we discussed the opportunities to offer the TRESPASS integrated process as a service to clients. Rather than focusing on our internal process architecture, this chapter started from the metaphor we developed in terms of security risk management as a navigation exercise, in which TRESPASS consultants can assist by means of the integrated tools. We believe that satellite images, maps, routes and road blocks make it easier to convey to clients the message of what our tools and processes do, and enable clear storytelling regarding the benefits of the methods.

This chapter provides a high-level service description for public use. A possible setup of the service model is represented in Table 12.1. Detailed exploitation plans are covered elsewhere. In such plans, specific details of the services, such as meetings with clients, may be adapted to the needs of the service provider and their market.

Stage	Consultants	Client	Meetings	Result
Satellite	Facilitate hands-on modelling	Engage in hands-on modelling	>= 8 hours	Shared (informal) system representation
Map	Formalise satellite view into TRE_S-PASS map	Optional: provide feedback	Optional: 2-4 hours	Navigator map and attacker profiles
Routes	Analyse scenarios	Evaluation of and feedback on routes/scenarios	Around 4 hours	Final routes/scenarios
Optimisation	Evaluate measures	Agree on measures	Around 2 hours	Effect of measures
Final report	Prepare report	Decide on actions	Around 4 hours	Final report

Table 12.1.: A possible setup of the service model

13. Conclusions

In this document we have presented the TRE$_S$PASS process, comprising methods and tools to model, analyse and visualise information security risks and countermeasures in dynamic organisations. These methods include data collection, risk modelling, formal analysis methods, social engineering and statistical research, as well as architectural and geographical aspects of security. These processes are supported by a suite of tools, to assist risk practitioners in identifying the most appropriate investment choices with regard to security expenditure and return on investment within their own organisations. The project regularly consults with information security and risk practitioners, in order to understand the types of tools which are best suited to their needs. A particular strength of the project has been the development of visualisations, which provide a more intuitive way of communicating the complexities of a given risk environment.

The diversity of the case studies, IPTV, Telco, Cloud and ATM, in terms of focus, scale and associated threats, has enabled the project to identify a number of key, core processes. These include both technical and social data gathering, a core TRE$_S$PASS model, which may be adapted or extended according to specific requirements, a process for attack generation and analysis and a suite of versatile visualisation processes which may be adapted according to the individual case. More bespoke processes, such as the e3fraud modelling approach to handle the commercial requirements of the Telco case study and visualisations of virtualised environments for the Cloud case study provide valuable extensions.

In addition to the case studies, the TRE$_S$PASS approach to process and tool development has been validated by means of a number of practitioner panels, held in different countries and with different practitioner communities. The practitioner panels were run using provocations and stimulus material developed within the project, including tools and prototypes. The feedback from these panels has provided valuable information for process development across the project.

In the last phase of the project, the developed TRE$_S$PASS process was validated against several established risk assessment frameworks.

The Estonian ISKE framework has provided a useful means of evaluating the structures within which practitioners might choose to employ TRE$_S$PASS processes and tools. While the TRE$_S$PASS approach differs from ISKE in some aspects, it would appear to have a complementary role to play in supporting information security risk practitioners. As ISKE is derived from the German BSI framework, which is also widely implemented, these findings may be relevant to a wider range of frameworks.

The CORAS risk assessment framework has by far the best thought-out client involvement process. This has served as one of the positive examples in the TRESPASS service model development.

We also compared TRESPASS to the FAIR risk taxonomy and assessment process. TRESPASS tools concentrate more on attacker's expected utility rather than the defender's loss, hence the two are not directly comparable. However, we feel that the TRESPASS approach complements FAIR in a positive way, since the rational attacker is actually more concerned about his own profit rather than the victim's loss.

TRESPASS made an important contribution to the TRICK risk assessment service developed by partner itrust. The process described in the current deliverable can be integrated into the TRICK Service workflow to select the optimal countermeasures to be implemented in an organisation. It also helps boosting the performance of the ADTop tool, and determining a set of optimal ISO/IEC 27002 security controls that have the largest added-value.

References

Baker, D. (2012, November). International revenue share fraud: Are we winning the battle against telecom pirates? *Black Swan Telecom Journal*. Available at: http://bswan.org/revenue_share_fraud.asp.

Bsi. (2013). Retrieved from https://www.bsi.bund.de/EN/Home/home_node.html

Cloud Security Alliance. (2010). *Top threats to cloud computing v1.0.* https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf.

CORAS. (2016). *http://coras.sourceforge.net/.*

De Gramatica, M., Labunets, K., Massacci, F., Paci, F., & Tedeschi, A. (2015). The role of catalogues of threats and security controls in security risk assessment: An empirical study with ATM professionals. In *Proc. of refsq* (Vol. 9013, pp. 98–114). Springer.

Dimkov, T., Pieters, W., & Hartel, P. (2010). 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. http://eprints.eemcs.utwente.nl/17295/.

Egelman, S., Brush, A. B., & Inkpen, K. M. (2008). Family accounts: a new paradigm for user accounts within the home environment. In *Proceedings of the 2008 acm conference on computer supported cooperative work* (pp. 669–678). New York, NY, USA: ACM. doi: 10.1145/1460563.1460666

ENISA. (2009). *Cloud Computing Risk Assessment* (Tech. Rep.). Author.

European-Commission. (2014). *Eurobarometer 79.4.* (Report). DG Communication. Retrieved from https://dbk.gesis.org/dbksearch/SDESC2.ASP?no=5852&db=e&search=&search2=&tab=&notabs=&nf=&af=&ll=

Gadyatskaya, O., Harpes, C., Mauw, S., Muller, C., & Muller, S. (2016). Bridging two worlds: Reconciling practical risk assessment methodologies with theory of attack trees. In *Proc. of gramsec.* Springer.

Gadyatskaya, O., Labunets, K., & Paci, F. (2016). Towards empirical evaluation of automated risk assessment methods. In *Proc. of crisis.* Springer.

Gordijn, J., & Akkermans, H. (2001). Designing and evaluating e-business models. *IEEE intelligent Systems*(4), 11–17.

Gordijn, J., Akkermans, H., Koks, A., & Schildwacht, J. (2004, April). *User manual e3-value editor.* http://e3value.few.vu.nl/docs/misc/manual_version2.pdf. Vrije Universiteit Amsterdam.

Group, T. O. (2009). *Technical Standard to Risk Taxonomy* (No. C081). http://pubs.opengroup.org/onlinepubs/9699919899/toc.pdf, accessed on 16.03.2013.

Hall, P., Coles-Kemp, L., & Heath, C. (2016). Visualisation in cyber-security: Towards a critical practice. In *Proc. of electronic visualisation and the arts austrolasia (evaa).*

Hay, B., Nance, K., & Bishop, M. (2011). Storm Clouds Rising: Security Challenges for IaaS Cloud Computing. In *Proceedings of the 2011 44th hawaii international conference on system sciences* (pp. 1–7). Washington, DC, USA: IEEE Computer Society.

Ionita, D., Hartel, P., Pieters, W., & Wieringa, R. (2013, September). *Current established risk assessment methodologies and tools* (No. TR-CTIT-14-04). Enschede, the Netherlands: University of Twente, Centre for Telematics and Information Technology (CTIT). Retrieved from http://doc.utwente.nl/89558/

Ionita, D., Wieringa, R., Wolos, L., Gordijn, J., & Pieters, W. (2015). Using value models for business risk analysis in e-service networks. In *The practice of enterprise modeling - 8th IFIP WG 8.1 working conference, poem 2015, valencia, spain, november 10-12, 2015. proceedings* (Vol. 235). Springer.

Iske. (2013). Retrieved from https://www.ria.ee/iske-en

ISO/IEC. (2011). *27005:2011 Information technology — Security techniques — Information security risk management.*

ISO/IEC. (2013). *27002:2013 Information technology — Security techniques — Code of practice for information security controls.*

Jones, J. A. (2005). *An Introduction to Factor Analysis of Information Risk (FAIR).* http://riskmanagementinsight.com/media/documents/FAIR_Introduction.pdf, accessed on 16.03.2013.

Labunets, K., Massacci, F., Paci, F., et al. (2013). An experimental comparison of two risk-based security methods. In *Proc. of esem* (pp. 163–172).

Labunets, K., Paci, F., Massacci, F., & Ruprai, R. (2014). An experiment on comparing textual vs. visual industrial methods for security risk assessment. In *Proc. of empire* (pp. 28–35). IEEE.

Li, E., Barendse, J., Brodbeck, F., & Tanner, A. (2016). From A to Z: Developing a visual vocabulary for information security threat visualisation. In *Proc. of gramsec.* Springer.

LLC, R. M. I. (2006). *Fair (factor analysis of information risk) basic risk assessment guide.* Author. http://www.riskmanagementinsight.com/media/docs/FAIR_brag.pdf.

LLC, R. M. I. (2010). *Fairlite high-level description.* Author. http://riskmanagementinsight.com/wp-content/uploads/2010/09/FAIRLite-Description-v2.pdf.

Lund, M. S., Solhaug, B., & Stølen, K. (2011). *Model-driven risk analysis - the CORAS approach.* Springer.

McEvoy, N. A., & Whitcombe, A. (2002). Structured risk analysis. In *Proceedings of the international conference on infrastructure security* (pp. 88–103). London, UK, UK: Springer-Verlag.

Mell, P., & Grance, T. (2009a, October). *Effectively and Securely Using the Cloud Computing Paradigm.*

Mell, P., & Grance, T. (2009b, October). *The NIST Definition of Cloud Computing.*

Moody, D. L. (2003). The method evaluation model: a theoretical model for validating information systems design methods. In *In proc. of ecis* (p. 1327-1336).

Muller, S., Harpes, C., & Muller, C. (2016). *Fast and optimal countermeasure selection for attack defence trees.* itrust consulting s.à r.l., University of Luxembourg, Telecom Bretagne. (Under scientific publication)

Pieters, W. (2011). Security and privacy in the clouds: a bird's eye view [Technical Report]. In S. Gutwirth, Y. Poullet, P. De Hert, & R. Leenes (Eds.), *Computers, privacy and data protection: an element of choice* (pp. 445–457). Dordrecht: Springer. http://eprints.eemcs.utwente.nl/19837/.

Pieters, W., Barendse, J., Ford, M., Heath, C., & Probst, C. (2016). The navigation metaphor in security economics. *IEEE Security & Privacy*.

Probst, C. W., & Hansen, R. R. (2008). An extensible analysable system model. *Information security technical report*, *13*(4), 235–246.

Probst, C. W., & Hunker, J. (2009). The risk of risk analysis—and its relation to the economics of insider threats. In *Proceedings of the 8^{th} annual workshop on the economics of information security (weis 2009).*

Refsdal, A., Solhaug, B., & Stølen, K. (2015). *Cyber-risk management*. Springer International Publishing.

Roy, A., Kim, D. S., & Trivedi, K. S. (2012). Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees. In *Proceedings of the 42nd annual ieee/ifip international conference on dependable systems and networks* (p. 299-310). IEEE.

The TRE$_S$PASS Project, D1.1.2. (2015). *Final specifications and requirements for socio-technical security models.* (Deliverable D1.1.2)

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

The TRE$_S$PASS Project, D1.3.4. (2016). *TRE$_S$PASS socio-technical security model and specification languages.* (Deliverable D1.3.4)

The TRE$_S$PASS Project, D2.2.1. (2013). *Technical data extraction prototype.* (Deliverable D2.2.1)

The TRE$_S$PASS Project, D2.2.2. (2015). *Data extraction from virtualized infrastructures.* (Deliverable D2.2.2)

The TRE$_S$PASS Project, D2.3.1. (2014). *Social data and policy extraction prototype.* (Deliverable D2.3.1)

The TRE$_S$PASS Project, D2.3.2. (2015). *TRE$_S$PASS social data and policy extraction techniques.* (Deliverable D2.3.2)

The TRE$_S$PASS Project, D2.4.1. (2016). *TRE$_S$PASS information system.* (Deliverable D2.4.1)

The TRE$_S$PASS Project, D3.2.1. (2015). *TRE$_S$PASS extraction methods for stochastic models.* (Deliverable D3.2.1)

The TRE$_S$PASS Project, D3.3.1. (2013). *First report on stochastic analysis methods.* (Deliverable D3.3.1)

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 TRESPASS Project, D3.4.2. (2016). *Methods for attack generation, preventive measures, and ranking.* (Deliverable D3.4.2)
The TRESPASS Project, D4.1.2. (2015). *Final requirements for visualisation processes and tools.* (Deliverable D4.1.2)
The TRESPASS Project, D4.2.1. (2014). *Initial report on visualizations of information security risks.* (Deliverable D4.2.1)
The TRESPASS Project, D4.2.2. (2016). *Methods for visualization of information security risks.* (Deliverable D4.2.2)
The TRESPASS Project, D4.3.1. (2014). *Initial visualisations of socio-technical dimensions of information-security risks.* (Deliverable D4.3.1)
The TRESPASS Project, D4.3.2. (2016). *Visualisations to simplify complex information.* (Deliverable D4.3.2)
The TRESPASS Project, D4.3.3. (2016). *Visualizations of socio-technical dimensions of information-security risks.* (Deliverable D4.3.3)
The TRESPASS Project, D5.1.2. (2015). *Final requirements for process integration.* (Deliverable D5.1.2)
The TRESPASS Project, D5.2.1. (2014). *Currently established risk-assessment methods.* (Deliverable D5.2.1)
The TRESPASS Project, D5.3.2. (2015). *Best practices for model creation and sharing.* (Deliverable D5.3.2)
The TRESPASS Project, D5.3.3. (2016). *Best practices for model maintenance.* (Deliverable D5.3.3)
The TRESPASS Project, D6.1.2. (2015). *Final requirements for tool integration.* (Deliverable D6.1.2)
The TRESPASS Project, D6.2.2. (2015). *Final refinement of functional requirements.* (Deliverable D6.2.2)
The TRESPASS Project, D6.3.1. (2015). *TRESPASS user interface.* (Deliverable D6.3.1)
The TRESPASS Project, D6.4.2. (2015). *TRESPASS tools handbook.* (Deliverable D6.4.2)
The TRESPASS Project, D7.1.1. (2013). *Initial requirements for implementation of case studies.* (Deliverable D7.1.1)
The TRESPASS Project, D7.2.2. (2016). *Final report case study a.* (Deliverable D7.2.2)
The TRESPASS Project, D7.3.1. (2014). *Results from case study b.* (Deliverable D7.3.1)
The TRESPASS Project, D7.3.2. (2016). *Final report case study b.* (Deliverable D7.3.2)
The TRESPASS Project, D7.4.1. (2014). *Results from case study c.* (Deliverable D7.4.1)
The TRESPASS Project, D7.4.2. (2016). *Final report case study c.* (Deliverable D7.4.2)

A. Process information leaflets

The project is working to create a range of process information leaflets, with the aim of providing a concise summary of each of the major processes being developed within the project. These are intended to provide an informative introduction to the process itself, an outline of the science associated with it, and links to resources where the reader can explore the subject in greater detail.

While these leaflets have been included in this process deliverable, they are also intended to be used to support case study partners in validating our processes and for dissemination purposes. Some examples of these leaflets are included in this section.

You can find us at:

https://www.trespass-project.eu

@TREsPASSproject

https://www.linkedin.com/company/trespass-project

References

[1] Jan-Willem H. Bullée, Lorena Montoya, Wolter Pieters, Marianne Junger, and Pieter H. Hartel (2015). The persuasion and security awareness experiment: reducing the success of social engineering attacks. *Journal of Experimental Criminology*, 11(1), 97–115. doi, 10.1007/s11292-014-9222-7

predict
prioritise
prevent

TRESPASS

This project receives funding from the European Commission's Seventh Framework Programme under Grant Agreement No. 318003 (TREsPASS)

predict
prioritise
prevent

TRESPASS

HOW TO: OUTWIT A SOCIAL ENGINEER

You have been travelling and just checked into your hotel room. As you walk into your room and set your bag down, your phone rings. A nice girl introduces herself as Rebecca from the front desk. She explains that there has been an issue during check-in and she needs to re-confirm your credit card information. Assuming she is calling from the hotel front desk, you provide your credit card information. She then informs you that everything has been resolved and wishes you a good stay.

Would you have given your credit card details?

predict
prioritise
prevent

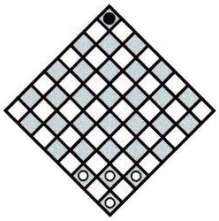
TRESPASS

DONT

- GIVE YOUR CREDENTIALS AWAY
  Your PIN & password belong to you, are yours only and are not meant to be shared. Once your PIN and password are shared with someone else, they know it as well. There is no such thing as returning a password without still knowing it.
- SAY YES TOO OFTEN
  People are helpful by nature, this is known by social engineers. The danger of social engineering is that you become a victim or contribute to an attack without knowing it.
- SHARE VALUABLE INFORMATION VIA A PHONE CALL
  The requests of social engineers tend to be harmless, but combining this information with other information can make it extremely dangerous. Is this piece of information they ask for really necessary for them to know?

DO

- VALIDATE IDENTITY
  It is known that social engineers adopt other people's identity to build trust with you. Try to figure out if they are who they claim to be.
- BE CRITICAL AND SUSPICIOUS
  The requests of a social engineer look legitimate and can be explained rationally. The hard part is to identify the difference between legitimate and illegitimate requests.
- ASK FOR NAME + PHONE NUMBER AND RE- DIAL
  Social Engineers can pretend to be your bank. To be sure they are the bank, check the number independently and dial them back.



Science bit

These hands-on tips, together with a short memo explaining what a social engineering attack is and a key chain, were tested in an experiment at the University of Twente. The aim of the attackers was to obtain the office keys of 118 university employees. Out of the 72 people that were not given advice in advance 45 (62.5%) gave their office key away to a stranger. Out of the 46 people that did get advice 17 (37.0%) gave their office key away to a stranger.

		Intervention		
		No	Yes	Total
Complied	No	27 (37.5%)	45 (62.5%)	72 (100%)
	Yes	29 (63.0%)	17 (37.0%)	46 (100%)
Total		56 (47.5%)	62 (52.5%)	118 (100%)

The full details of this experiment can be found in our article in the Journal of Experimental Criminology [1].

You can find us at:

https://www.trespass-project.eu

#TREsPASSproject

https://www.linkedin.com/company/trespass-project

References

[1] Paper/further info. reference.

predict
prioritise
prevent

TREsPASS

This project receives funding from the European Commission's Seventh Framework Programme under Grant Agreement No. 318003 (TREsPASS)

HOW TO: MODEL ATTACKS USING TIMED AUTOMATA

Security practitioners often express the desire for simpler, yet more powerful risk management processes. In response to these expectations, the TREsPASS project has been exploring the use of timed automata for risk modeling, attack generation and analysis.

If you're looking for a flexible way of modeling and analyzing a scenario, with options for incorporating different parameters such as time and cost, timed automata might be the right choice for you.

STEPS IN THE PROCESS

- CREATE A MODEL
  This may be hand crafted, developed using your preferred modeling technique, or derived from intermediate structures such as attack trees.
- SELECT PARAMETERS
  There are numerous different parameters which can be incorporated into the timed automata process. Often impact is of primary concern within an organisation, however elements such as the cost, time and skill level required to perform specific steps may inform the choices of an attacker.
- RUN THE ANALYSIS
  Model checking and related techniques can be used for finding new attacks, as well as many other kinds of analysis that are useful for evaluating the security of a system, e.g. the potential impact of an attack or the time it takes to perform an attack.
- REPEAT

Adjusting the parameters of the analysis allows the scenario to be evaluated from multiple different perspectives. It also allows for the incorporation of extra elements such as countermeasures. Comparison of successive outcomes provides information as to the relative effectiveness of different interventions.

RESULTS

- COMPLETE LIST OF POSSIBLE ATTACKS
  Using timed automata allows an intuitive modeling of a system, in which quantities like time and cost can be easily added. Thanks to automated model checking and automata theory, we can automatically generate possible attacks in our model.
- ATTACK ANALYSIS AND SIMULATION

  Automated model checking and automata theory further supports the analysis and simulation of both model and attack, revealing details about the specific interaction between the attacker and their target.
- SCENARIO COMPARISON
  The analysis can be rerun with different parameters, allowing for different outcomes to be assessed according to the scenario selected.

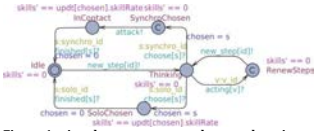


Figure 1 - timed automata example - attacker view

Science bit

As part of the TREsPASS project, Aalborg University has been using its UPPAAL tool set to develop novel approaches to risk modeling using timed automata.

Timed automata offer considerable versatility, making them a highly adaptable option when developing organisational security risk models. In addition, they can support attack generation, incorporating values such as time and cost. They are also compatible with other tried and tested approaches, such as attack trees. In the context of attack trees, they provide an extension to the standard functionality, supporting the evaluation of additional factors such as countermeasures.

The advantage of having formal models of a system, is that it enables the full range of formal methods to be used in modelling, analysing, and verifying the (in-)security of a system.

Further details of the TREsPASS approach to modeling timed automata can be found in our article in xxxx [1].

The UPPAAL tool set is freely available for download from http://www.uppaal.org/.

You can find us at:

https://www.trespass-project.eu

@TREsPASSproject

https://www.linkedin.com/company/trespass-project

References

[1] 'Your source paper' ref. e.g. Jan-Willem H. Bullée, Lorena Montoya, Wolter Pieters, Marianne Junger, and Pieter H. Hartel. The persuasion and security awareness experiment: reducing the success of social engineering attacks. *Journal of Experimental Criminology*, pages 1–19, 2015.

predict
prioritise
prevent

TREsPASS

This project receives funding from the European Commission's Seventh Framework Programme under Grant Agreement No. 318003 (TREsPASS)

predict
prioritise
prevent

TREsPASS

HOW TO: MAKE A TREsPASS 3-FOLD

Which aspect of TREsPASS do you find most interesting? Could you provide a brief introduction to it?

How might a security practitioner use it when working with clients?

What actions should they take? What actions should they avoid? What degree of effort will be required?

predict
prioritise
prevent

TREsPASS

DONT

- OVERFILL
  If you can't fit everything in, you may want to think again about your audience. They can always go to the reference(s) if they want more.
- OVERSIMPLIFY
  The reader needs to be able to get a reasonable feel for what is being described. Check with a colleague if you're not sure.

DO

- KEEP IT LIGHT
  You're writing for busy practitioners.
- COVER ALL THE GROUND
  The description should be brief but essentially complete. Keep the detail light, but don't miss out anything major.
- PROVIDE A REFERENCE
  Make it clear where the reader goes next if they are inspired to investigate further.
- INCLUDE A PICTURE
  A simple diagram can add interest and provide context.

Science bit

OK, not much science in this case, although RFC 2223 has served as inspiration.

Regarding levels of abstraction, sometimes it is easier for a non-expert to write a brief guide like this. However, it should always be checked with someone who has the necessary expertise.

The aim is to build up a library of leaflets to support people in using the outputs from TREsPASS.

The full details of this experiment can be found in our article in 'Your source paper' [1].