predict
prioritise
prevent

TRESPASS

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

Deliverable D6.4.3

TRESPASS deployment and maintenance plan

Project:	TRESPASS
Project Number:	ICT-318003
Deliverable:	D6.4.3
Title:	TRESPASS deployment and maintenance plan
Version:	1.0
Confidentiality:	Public
Editor:	Cédric Muller
Cont. Authors:	F. Arnold, L. Coles-Kemp, C. W. Probst, A. Lenin, J. Willemson, F. Reis, E. Martin, M. Martins, A. McKinnon, B. Jager, C. 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
ITR	Miguel Martins, Alex McKinnon, Benoît Jager, Cédric Muller	ALL
UT	Florian Arnold	5
RHUL	Lizzie Coles-Kemp	5
DTU	Christian W. Probst	5
CYB	Aleksandr Lenin, Jan Willemson	5
GMVP	Fátima Reis, Enrique Martin	5

Quality assurance		
Role	Name	Date
Editor	Cédric Muller (ITR)	2016-10-31
Reviewer	Aleksandr Lenin (CBY)	2016-10-12
Reviewer	Sjouke Mauw (UL)	2016-10-21
WP leader	Carlo Harpes (ITR)	2016-10-31
Coordinator	Pieter Hartel (UT)	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 (TRESPASS). This publication reflects only the authors' views and the Union is not liable for any use that may be made of the information contained herein.*

Contents

List of Figures

List of Tables

Management summary

Key takeaways:

- This document describes how the TRESPASS platform works and what are the technical features. It contains guidelines in order to know how to deploy and maintain the platform, as well as a review of the TRESPASS tools handbook, which briefly describes the tools and explains how to integrate them to the platform.
- This deliverable presents the integrated TRESPASS tools. The integration platform has a graphical user interface and several tools have been integrated into it. Tools are using general computational procedures and approaches that can be applied to any domain, but some components, like the Attack Pattern Library (APL), or the Treemaker contain domain-specific information about the TRESPASS case studies.

At the end of the project, the following tools are available:

- Architect, a tool used to build models of infrastructures, processes, etc.;
- Attack generation tools that are able to create attack trees from model files: TRESPASS Model and Treemaker;
- ADTool, which can be used for the construction, manipulation and visualisation of attack trees;
- Attack Pattern Library (APL) increases granularity of the models, pushing analysis beyond the limits of the TRESPASS model by populating the automatically generated attack tree with domain-specific and reusable components. It is used to promote the reuse of modular elements to improve the process of model development. The APL is also used to decorate the attack trees with parameters that are needed in the analysis;
- Analytical tools:
  - The Attack-Defence Tree optimiser (ADTop), which can be used to improve security risk treatment by finding optimal security controls in attack trees based on cost-benefit analysis;
  - The Attack Tree Analyser (ATA), which can be used to compute optimal attack vectors, taking the attacker profile into account;
  - The Attack Tree Evaluator (ATE), which can be used to compute the Pareto set of pairs with maximum probability and minimum cost, if costs are provided;
  - e3fraud, which is a tool for fraud analysis;

- ATtop, which can be used to interchange attack trees between different tools by model transformation, or to analyse expected cost and time of attacks;
- TRICK Service, which can be used to conduct risk assessments according to ISO/IEC 27005 or CSSF 12/544;
- ArgueSecure, which consists of a collaborative web-based tool and a Java tool for offline use, in order to support the documention of the structured arguments and defensibility relationships elicited as part of an informal qualitative risk assessment.

Some of these tools are still being improved by TRE$_S$PASS partners, and are continuously integrated in the TRE$_S$PASS platform as soon as new versions become available.

The TRE$_S$PASS platform is intended to be published as open-source at the conclusion of the project.

1. Introduction

This report describes the integration platform of the TRESPASS tools. After the conclusion of Task T6.3: "The TRESPASS user interface", a new, state-of-the-art user interface to TRESPASS tools is available, which conforms with industry best practices. The web-based graphical interface of the integrated tools is accessible from any web browser.

The integrated TRESPASS tools are hosted at https://trespass.itrust.lu, which also contains a user interface intended mainly for administration purposes.

The TRESPASS user interface is available on:
https://trespass.itrust.lu/attack-navigator/index.html.

In order to access the tools from the TRESPASS user interface, an API has been developed. It is described on https://trespass.itrust.lu/api/json.

The prototype is then composed of a user interface, a platform and a set of tools that run on that platform. Some tools interact with others through the concept of toolchains (chains of tools).

1.1. Goals of the tool handbook

This document includes the revised sections from the tool handbook, which serves the following purposes:

- Manual for integration: the integration of new tools, both on the user interface and on the software server, is described;
- Manual for deployment: on the one hand we describe how tools are made available for internal purposes and, on the other hand, how the entire platform is used in consultancy missions after the project;
- Manual for maintenance: the maintenance of the platform consists of keeping up-to-date software packages, applying bug fixes to the platform and individual tools, and the execution of functional tests after maintenance operations.

1.2. Foreground and background

Early versions of some TRESPASS tools (ADTool (v1.1), DFTCalc, ApproxTree+) are background of the deliverable. For Architect, the generic configurable modelling platform, as well as the ArchiMate modelling tool are background. The risk and security modelling and analysis extension are part of the foreground, and the specific configuration for editing TRESPASS (infrastructure and process) models is foreground. The other integrated tools have been developed or improved by their development teams and constitute foreground of the project. Some tools like TRICK Service can be a mix of background, foreground, and sideground. Everything else in this document and on the prototype websites is foreground of WP6 and includes mainly the configuration of servers and operating systems, the implementation of both user interfaces, and the integration of the available tools and additional supporting features.

1.3. Structure of the document

The remaining sections of this document define the scope and the objectives of the prototype (chapter 2), and specify the prototype itself (chapter 3), namely the physical architecture, tools that have been included and the web interfaces.

Chapter 4 is a guide for the TRESPASS platform and chapters 5, 6, and 7 describe the integration, deployment, and maintenance of the TRESPASS platform respectively.

Chapter 8 concludes the document.

Appendix A provides the context for this deliverable in the TRESPASS project. It describes the overall summary of the project and the TRESPASS workflow.

Appendix B provides details about the WP6 targeted requirements.

Appendix C provides the list of final tools that are being developed in the context of the TRESPASS project.

1.4. Acronyms and abbreviations

Symbol	Explanation
ADTool	Attack-Defense Tree Tool
ADTop	Attack-Defence Tree optimiser
ADTree	Attack-Defence Tree
AN	Attack Navigator
ANM	Attack Navigator Map
API	Applications Programming Interface
APL	Attack Pattern Library
ATA	Attack Tree Analyzer

Symbol	Explanation
ATCalc	Attack Tree Calculator
ATE	Attack Tree Evaluator
ATM	Automated Teller Machine, an electronic banking outlet
ATree	Attack Tree
ATREES	Attack Trees
ATtop	Attack Tree tool optimiser
CADP	Construction and Analysis of Distributed Processes
CAPEC	Common Attack Pattern Enumeration and Classification dictionary
CAPTCHA	Completely Automated Public Turing Test To Tell Computers and Humans Apart
CAS	Central Authentication Service
CEAV	Cloud Environment Actor Visualiser
CherryPy	Object-oriented web application framework using the Python programming language
CPU	Central Processing Unit
CSV	Comma-separated values
CVE	Common Vulnerabilities and Exposures
DB	Database
DDR3	Double Data Rate 3rd generation memory
DFTCalc	Dynamic Fault Tree Calculator
DNS	Domain Name System
DoW	Description of Work
ECMA	European Computer Manufacturers Association
FE	Front-End
GitHub	Web-based Git repository hosting service
GB	GigaByte
GHz	GigaHertz
GUI	Graphical User Interface
HD	Hard Drive
HTTP(S)	HyperText Transfer Protocol (Secure)
IBM	International Business Machines
INRIA	Institut national de recherche en informatique et en automatique
IPTV	Internet Protocol Television
ISKE	Intelligent System and Knowledge Engineering (IEEE International Conference)
ISMS	Information Security Management System
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
JAR	Java Archive, a package file format typically used to aggregate many Java class files
JDK	Java Development Kit
JNLP	Java Network Launching Protocol
JSON	JavaScript Object Notation
KVM	Kernel-based Virtual Machine
LDAP	Lightweight Directory Access Protocol
LTS	Long Term Support
Mac	Macintosh (slang for Apple computer)
MRD	Macintosh Remote Desktop, now Apple Remote Desktop (ARD)
MT	Management Team
MTTF	Mean Time To Failure
MySQL	Open source relational database management system based on the structure query language (SQL)
N/A	Not Applicable
NIST	National Institute of Standards and Technology
OS	Operating System
PC	Personal Computer

Symbol	Explanation
PDF/SVG	Portable Document Format/Scalable Vector Graphics
R	Requirement
RAM	Random Access Memory
RDF	Resource Description Framework
RDP	Remote Desktop Protocol
ROSI	Return On Security Investment
SAVE	Security Audits of heterogeneous Virtual Environments
SC	Software Checker tool
SCALA	Scalable Language
SSL	Secure Socket Layer
SVN	Apache Subversion tool
SVRS	Security Vulnerability Repository Service (EU project)
T	Task
TBD	To Be Defined
TCS	TRE$_S$PASS Core System
TiCoVis	Time-Containment Visualiser
TKB	TRE$_S$PASS Knowledge Base
TLS	Transport Layer Security
TM	Trade Mark
TOGAF	The Open Group Architecture Framework
TREsPASS	Technology-supported Risk Estimation by Predictive Assessment of Socio-technical Security
TRICK	Tool for Risk management of an ISMS based on a Central Knowledge base
TRL	Technology Readiness Level
TS	TRICK Service web application
U	Use-case
UPPAAL	Acronym based on a combination of UPPsala and AALborg universities
URL	Uniform Resource Locator (a protocol for specifying addresses on the Internet)
V	Version
VM	Virtual Machine
WP	Work Package
XLS	eXceL Spreadsheet, Microsoft Excel file extension
XML	eXtensible Markup Language

Table 1.1.: List of acronyms.

2. Prototype scope and objectives

According to the DoW, task T6.4 is responsible for deployment and maintenance of the TRESPASS tools. Up to now, different tools have been adapted or developed by other work packages and most of them have been integrated on the platform in the scope of this task.

The prototype of the TRESPASS tools includes the final implementation of the TRESPASS tools from WP1 to WP5 and their initial integration in a common platform and a graphical user interface. The final level of integration allows users to provide input to a tool and download the output (generally text files).

The output of a tool can be fed to the next tool as input. This process has been automated with the creation of toolchains and only the input for the first tool needs to be provided.

The implementation of the final platform took place in parallel with the refinement of functional requirements (The TRESPASS Project, D6.2.2, 2015). For the final version of the platform and of the integrated tools, the relevant requirements have been considered, namely (The TRESPASS Project, D6.1.2, 2015). The entire set of requirements is also considered here, in this final version of the deliverable.

The final TRESPASS platform contains a set of tools to be used by an operator in the client's premises, with real data from the organisation. That data is likely to be confidential and according to security policies the operators might not be allowed to send data to an external server, or even to connect to a network cable.

In order to allow all project partners to develop, test and use the tools, the prototype has been developed with web interfaces and runs on servers at itrust consulting premises. At the end of the project, the final set of tools can be packed in a virtual machine and deployed on any PC and be used without any network connections. For this to be possible, we prefer the integration of tools that can be fully installed on the tools server, without accessing any external servers. Tools that can be accessed remotely are also allowed but the consortium is aware of the fact that their use might not be allowed in most of the situations.

3. Prototype specification

We have decided in (The TRESPASS Project, D6.1.2, 2015) to adopt a loosely coupled solution, using a centralised integration component and database. The modules coming from the different work packages in the scope of the prototype exchange data either between tools directly or through the central component.

The tools of the prototype were developed or adapted focusing on the project case studies.

3.1. Physical architecture

This section describes the physical architecture of the servers running TRESPASS tools.

We expect that tools run on different operating systems or require specific appliances, so we decided to segregate the execution of the different services on different dockers.

Because of the limited number of users, we expect that the same physical machine is able to run the final integration platform. The physical machine has the following specifications:

- Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
- RAM: DDR3 10 GB
- HD: 20 GB
- OS: Ubuntu Server 16.04.1 LTS
- Additional HD for data and programs: 20 GB

3.2. TRESPASS user interfaces

To access the prototype, users only need a web browser and to type the following address `https://trespass.itrust.lu/attack-navigator/index.html`. They need first to login to the TRESPASS platform `https://trespass.itrust.lu`, on which they can also try the tools individually.

3.2.1. The Attack Navigator

The Attack Navigator is the TRESPASS user interface. It is described in (The TRESPASS Project, D6.3.1, 2015). Here we focus on basic access options of the user interface.

The home screen of the Attack Navigator is shown in Figure 3.1.

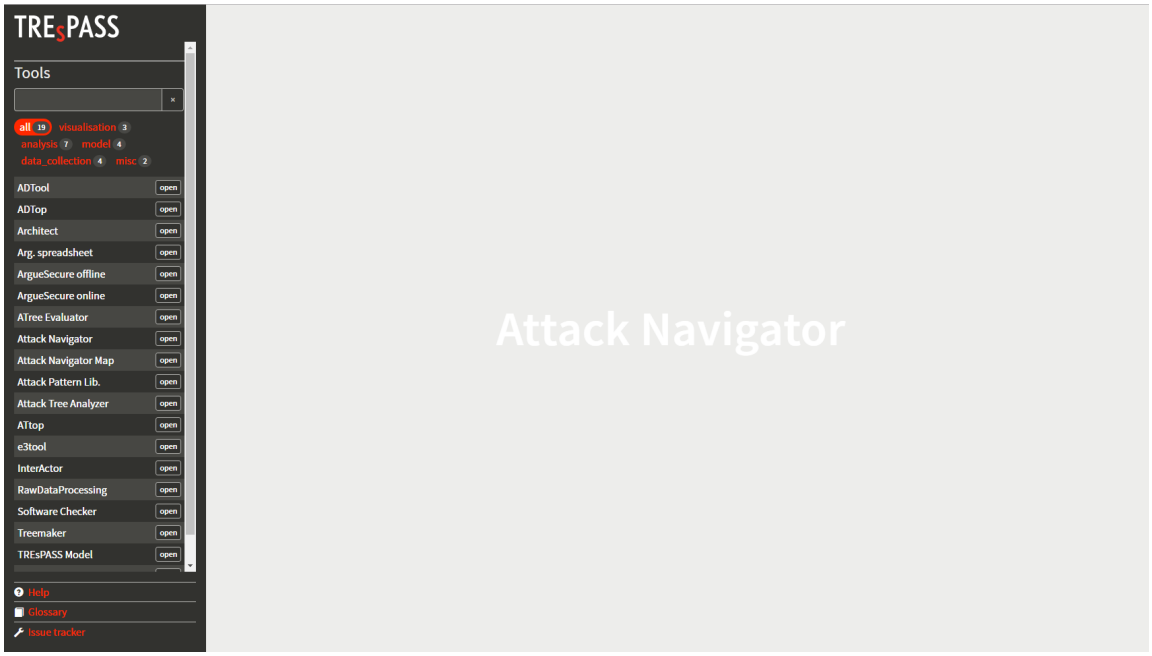

Figure 3.1.: AN home page.

Whenever the user runs a tool, a small description of the tool is presented and the form to upload its input file is shown, as in Figure 3.2

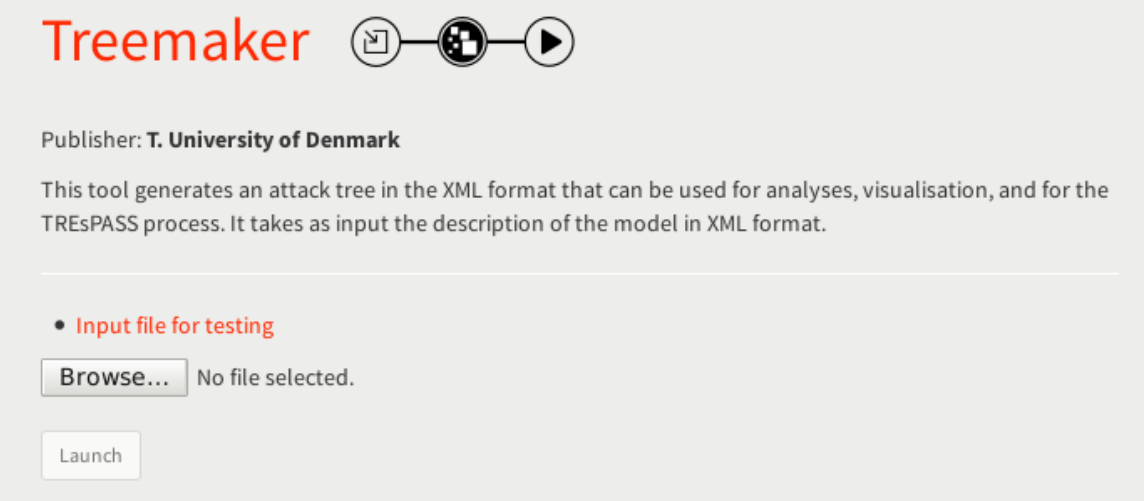

Figure 3.2.: Example of a tool (Treemaker).

3.2.2. The TRESPASS platform

The TRESPASS platform is an environment where the tools developed within the project can be viewed and executed. The platform is available 24/7, except during short maintenance periods. For the detailed description of each tool, its capabilities and limitations, the user is advised to refer to the documentation produced by the respective work package/publisher.

The platform and most of its tools are not mature enough to be used in a real scenario with huge sets of data or requiring significant processing power. The platform shall only be used for the prototypes in the scope of the project.

The following tools, grouped by publisher and in no specific order, are included in the platform:

Lust:

- AN: the Attack Navigator is the TRESPASS user interface, and it is complementary to the TRESPASS platform for tools. The Attack Navigator user interface has to be seen as an entry point. It is an environment where all tools developed within the TRESPASS project can be viewed, accessed and connected, in order to help support prediction, prioritisation and prevention of complex misuse scenarios. Next to access to tools it provides a help function and a glossary of terms used in the project.
- ANM: the Attack Navigator Map 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, with assets, items, actors,

processes, policies and a set of attacker profiles. The analysis of the scenario is visualised to help make decisions.

itrust consulting:

- TRICK Service: TRICK Service assesses quantitative risks and fulfils risk treatment operations according to the user-configured referential standards (ISO 27001, 27002, etc.). It outputs the risk treatment plan which prioritises the implementation of security measures according to their Return On Security Investment (ROSI), risk specificities and feasibility.
- ADTop: ADTop (Attack-Defence Tree optimiser) is a tool for risk analysis which takes an attack tree and an extract of risk analysis as input information and 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).
- Software Checker: determines if specific software versions contain any public vulnerability indexed by reliable databases such as CVE (http://nvd.nist.gov/).

Cybernetica:

- Attack Tree Analyzer: the attack tree computation tool (ATA) can be used to calculate optimal attack vectors (from the attacker point of view) taking attacker profile into account. ATA analyses if the considered threat model is feasible to rational profit-oriented adversaries. In the case of positive answer, it provides the user with the list of top-10 most feasible attack vectors. Converter and Failure-free Model were some time ago stand-alone tools, but they are now part of the ATA tool.
- Attack Pattern Library (APL): the Attack Pattern Library (APL) is intended to promote the reuse of modular elements to improve the process of model development. It decorates Treemaker output with attack patterns, as well as attack tree leafs with quantitative annotations.

University of Luxembourg:

- ADTool: the Attack-Defense Tree Tool (ADTool) allows users to model and analyse attack-defense scenarios represented with attack-tree and attack-defense trees models. It supports the methodology developed within the ATREES project. It implements many relevant attributes (attribute domains) for quantitative analysis, such as cost, satisfiability of a scenario, probability or an attack, and new attributes can be defined in the tool. The bottom-up computation algorithm is used for evaluating the selected attribute for the whole tree. The current stable release of the ADTool is able to import automatically generated attack trees from the APL, show top attack vectors in a tree (for some attributes), and it can be run in the scripting mode.

Technical University of Denmark:

- TRESPASS model: the TRESPASS model provides an abstraction layer for the data store. It is currently mainly used as interface (API) between the information stored in the XML file and Treemaker, as well as projects working directly on the model.
- Treemaker: Treemaker identifies possible attacks in the model to reach an attack goal and generates attack trees (in XML format) for these attacks that can be used for analyses, visualisation, and for the TRESPASS process. It takes as input the description of the model in XML format. The approach to attack generation is based on policy invalidation on the socio-technical security model, and identifies ways of breaking policies in a system through recursive refinement of goals and identification of missing assets. The sequence of actions to identify missing assets and the actual actions performed is then translated into an attack tree.
- ATree Evaluator: the ATree Evaluator (ATE) computes the Pareto set of pairs with maximum probability and minimum cost, if costs are provided. The tool addresses multi-parameter optimisation of attack trees in terms of Pareto efficiency. The evaluation techniques characterise the leaves of a tree with more than one parameter, such as success probability and cost of an attack. Such multi-parameter optimisations are necessary in case of conflicting parameters, as there is no single best solution but rather a set of optimal solutions. We handle conflicting parameters by computing the set of efficient solutions, defined in terms of Pareto efficiency. The results computed by the ATE allow the defender for example to identify the maximum probability for a successful attack, given an attacker budget.

GMV Portugal:

- RawDataProcessing: the Raw Data Converter tool is composed of a number of modules. The goal of these modules is to become the root of a general tool which converts raw data from several data sources into Trespass input data (WP1 input data).

University of Twente:

- ATtop: it has the same functionality as originally foreseen for the Attack Tree Calculator (ATCalc). In other words, ATtop, which is inheriting the feature from ATCalc to communicate with several tools, can automatically generate an input file for ATCalc. ATtop provides two functionalities. It can first input and output a variety of attack tree dialects; so it can be used to interchange attack trees between different tools by model transformation. Second, depending on appropriate quantitative values in the basic attack steps, it can analyse expected cost and time of attacks. It can compute expected cost under a time bound, expected time under a cost bound, or display the cumulative risk (probability x impact) over time. ATtop transforms attack trees to Priced Timed Automata, reusing UPPAAL as the underlying analysis engine.
- ATCalc: this tool is not directly integrated in the TRESPASS platform, but it has to be mentioned, because it was the former tool developed for efficient attack tree analysis. ATCalc computes the system unreliability for each mission time, i.e. the probability that the system fails within the mission time, as well as the Mean Time To Failure (MTTF), i.e. the expected time that the system will fail. ATCalc uses CADP, proprietary software owned by INRIA.

- e3tool: e3tool supports the construction of e3value value models as well as conducting profitability and fraud assessment of these models, based on the e3fraud methodology. An e3value model describes how economic value is created and exchanged within a network of actors. The e3fraud methodology consists of an ontological extension to e3value able to represent "sub-ideal" (i.e. fraudulent) value models, and a generation module which can identify, rank and visualise fraudulent variations of a given value model.
- RDFparser: this tool is not directly integrated in the TRESPASS platform, but it has to be mentioned too, because it was a small Java tool which acted as an extension to the e3value toolkit. It took as input (.rdf) e3value models and output profitability graphs based on selected parameters. RDFparser was the first version of what is now e3tool.
- ArgueSecure: it is a set of argumentation-based risk assessment tools classified into the “Data collection” category, which aim at documenting the rationale behind attacks identified and countermeasures selected as part of an informal, qualitative risk assessment. It has three different versions: a lightweight Excel-based version, a Java version which is intended to be used during dedicated security requirements elicitation sessions and designed to be usable with a projector, and an online version which in addition allows stakeholders and experts to engage in a risk assessment in real-time without being in the same room and even without being available at the same time. The first version of the tool was Arg. Spreadsheets, an Excel-based tool which supports documenting the structured arguments and defensibility relationships elicited as part of an informal argumentation game. It was recursively computing argument states and tags arguments with the components or assets they refer to.

BiZZdesign:

- Architect: Architect is a leading software tool for Enterprise Architecture. Architect is compliant with ArchiMate and TOGAF, the open standards for Enterprise Architecture, maintained by The Open Group. The access to Architect is through RDP (natively supported on Windows, MRD for Mac, Remmina on Linux).

Royal Holloway University of London:

- InterActor: this tool is classified in the “Data collection” category. Users create private projects where initial problem-definitions are stated. Data from any number of separate workshops (including physical modelling) can be entered manually or imported as .csv or .json files. Using network graphs and a flow chart view the user can investigate and order the data, and construct a narrative from it for use with other tools.

IBM Research: The following tools are not directly integrated in the TRESPASS platform, but they need to be mentioned, because they work as separate underlying tools for the ANM.

- SAVE: this tool is an IBM tool that automates the extraction of infrastructure details from virtualised environments. The tool supports a number of cloud technologies

such as VMWare, OpenStack, KVM. The data is stored in an abstracted format that allows the formal modelling of isolation policies and properties. Data extraction creates files in the TRESPASS XML format and can be loaded into the ANM.

- VMWare Extract: this tool extracts technical information from VMWare networks. The information includes infrastructure details and includes access control policies. The tool stores data in the TRESPASS defined format.
- Knowledge Base: it is used by visualisation and analysis tools and serves, as the TRESPASS Information System, to gather all related input and configuration files that are required during the creation of a TRESPASS model for a given scenario and the run of analysis tools. It supports version control of all configuration and generated files, allowing a complete capture of the work on a specific scenario during all changes and re-runs. It is mainly used as the backend for the running of the ANM.

4. Guide for the TRESPASS platform

4.1. Registration

In order to access the TRESPASS platform or the user interface, users need to register to the following address: https://trespass.itrust.lu. They are then redirected to the Central Authentication Service (CAS) https://trespass-cas.itrust.lu authentication page, as shown in Figure 4.1. More details about the authentication process are given in section 7.4.1.

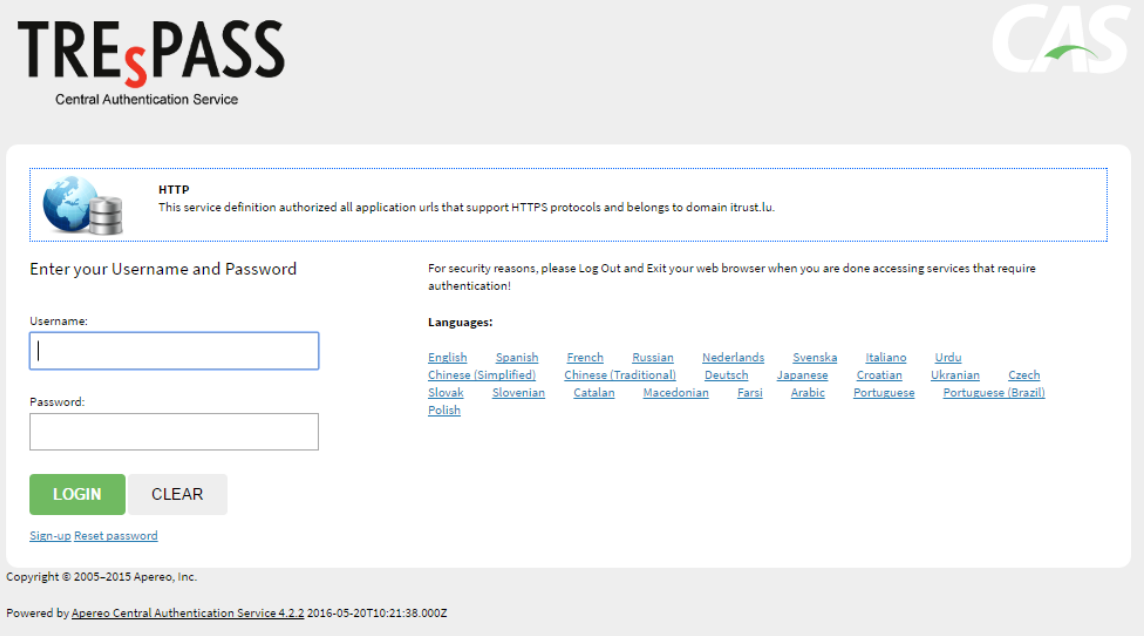

Figure 4.1.: TRESPASS login page.

In this page, users have to click on the "Sign-up" link located at the bottom of the page below the LOGIN button. They are then redirected to the registration page, as showed in Figure 4.2.

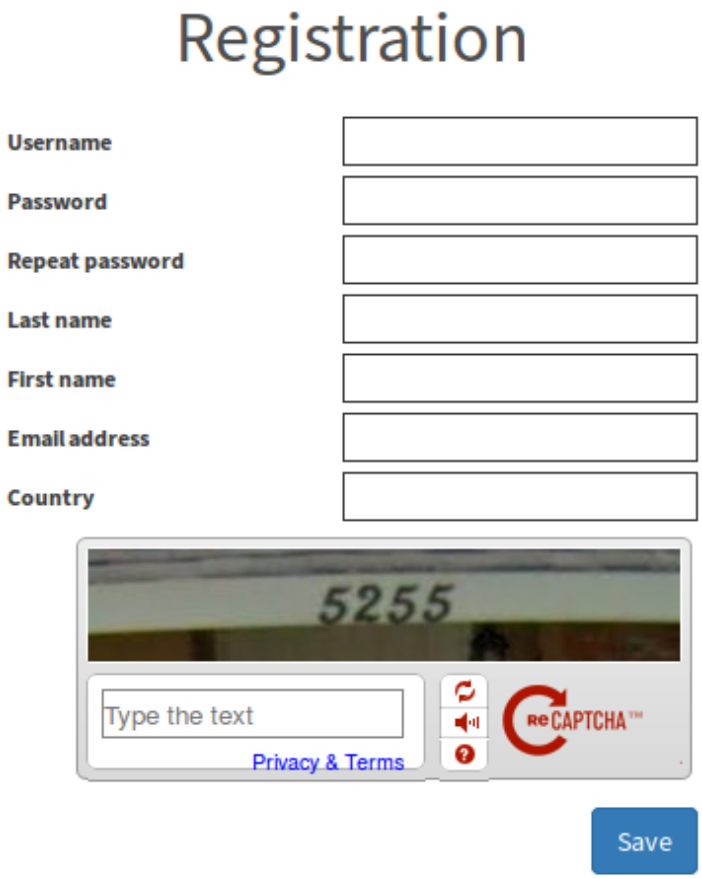

Figure 4.2.: Registration page.

All the fields shown in Figure 4.2 must be correctly completed in order to create an account. Once users have filled in all these fields, including CAPTCHA, they should click on the "Save" button to create an account.

As soon the registration page is posted, users receive an email containing a web link to open in order to validate the provided email address. To complete the registration process, users must click on this link, and the administrator must manually validate the account. Finally, users receive a confirmation email that the account has been validated, and they can enter their credentials to access the TRE$_S$PASS platform.

This restrictive registration process avoids automatic creation of accounts by bots, and enables to supervise registered users. In particular, users need to be involved in the TRE$_S$PASS project for confidentiality reasons with regards to some specific applications. Security elements are described in more details in Section 4.5.4.

After a successful login, the home page is displayed, as shown in Figure 4.3.

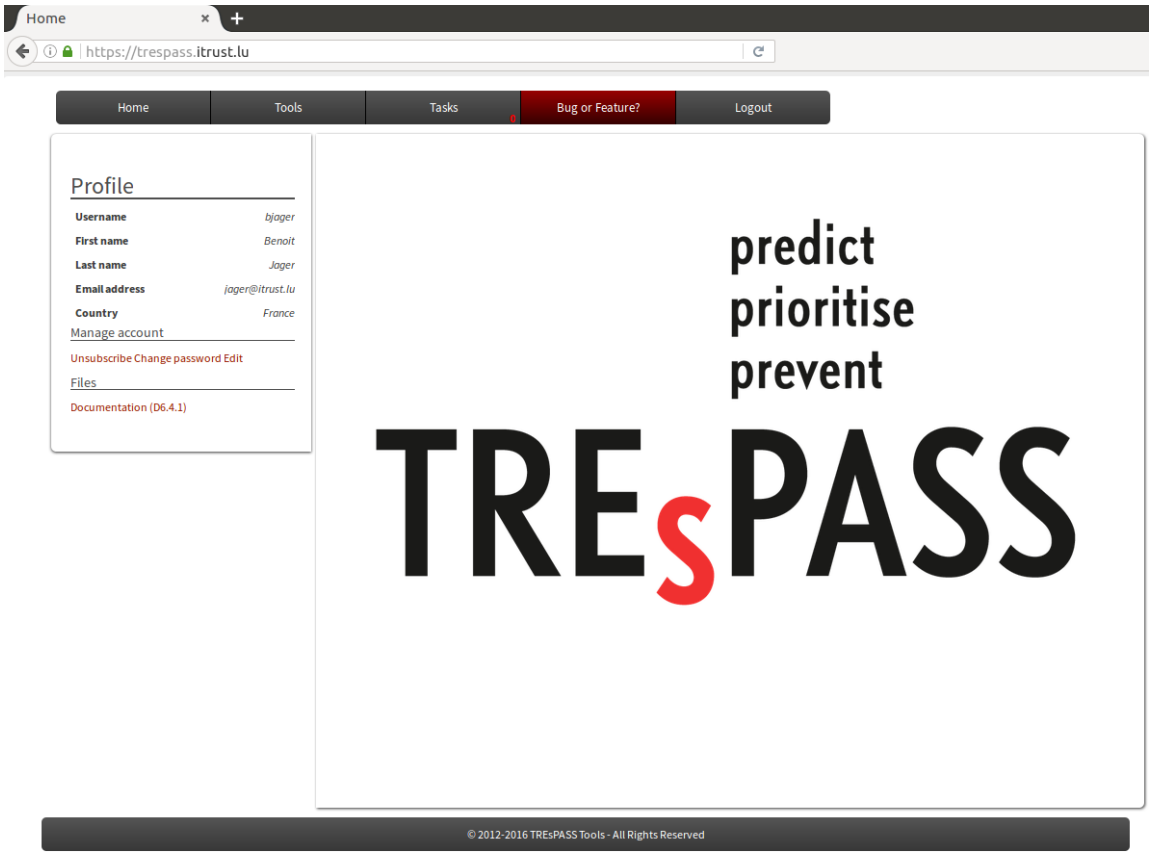

Figure 4.3.: TRE$_S$PASS platform home page.

4.2. The provided services

Following services are provided by the TRE$_S$PASS platform:

- https://trespass.itrust.lu: it is the main application which provides features to add and manage tools;
- https://trespass-cas.itrust.lu: it is the authentication service, providing single-sign on in order to authenticate once on one service and being able to access other services without authenticating again;
- https://trespass-svn.itrust.lu: it provides subversion repository access but is not using the CAS server because of compatibility issues with SVN clients. The authentication is made through the LDAP server, so same credentials than for CAS authentication can be used.
- https://trespass.itrust.lu/attack-navigator/: it is the entry point to the TRE$_S$PASS user interface;

- https://trespass.itrust.lu/attack-navigator-map/: it provides Attack Navigator Map and Knowledge Base tools;
- https://trespass-ticket.itrust.lu/: it provides a ticketing system in order to post issues for bug and support on tools.

4.3. Migration of the TRESPASS platform to a virtual machine

The TRESPASS platform is envisaged as a set of tools to be used by an operator in client's premises, with real data from the organisation. This data is likely to be confidential and according to security policies operators are not allowed to send data to an external server, or even to connect a network cable. Based on this, most of the tools can be packed in a virtual machine that is able to run on a laptop and be used without any network connection. Tools that are hosted on external servers cannot be integrated. Its use in a real scenario is subject to a special authorisation by the organisation that is being investigated. The virtual machine will be provided at the end of the TRESPASS project, and the TRESPASS platform will be published as open-source.

Update (2017-04-21): The latest version of the TRESPASS virtual machine and related documentation is available at:
https://github.com/itrust-consulting/trespass-vm-enduser.

4.4. Configuration

4.4.1. Default user access

When authenticated to the https://trespass.itrust.lu website as default user, the user has access to the following entries in the menu:

- **Tools**, which links to the list of tools integrated in the website;
- **Tasks**, which links to the tasks the user launched;
- **Bug or feature?** which links to the https://trespass-ticket.itrust.lu website to report on bugs and features (redirection to https://trespass-redmine.itrust.lu).
- **Logout**, which logouts user from all services.

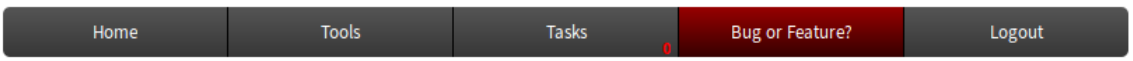

Figure 4.4.: Default user access.

4.4.2. Administrator access

When authenticated as an administrator, on top of default user access menu entries, the user has access to:

- **Manage users** which provides features to validate and delete users;
- **Manage tools** which provides features to add, edit and remove tools from the website.

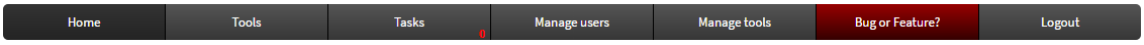

Figure 4.5.: Administrator access.

4.4.3. Manage users

This page allows administrators to manage the registered users of the TRE$_S$PASS platform. Here, administrators are able to see the details of all users who are currently registered, modify their access rights or even remove (delete) user accounts. For each user, the following information is given:

- The **Full name** of the user;
- The **Status** of the user, i.e. whether the account is enabled or not;
- The **Roles** that the user has, i.e. User or Administrator;
- The **Manage Roles** section, where administrators can modify the roles given to the user.

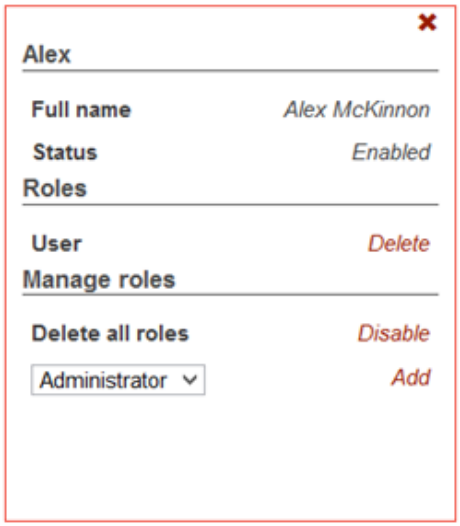

Figure 4.6.: Example of a user.

Users can be assigned the following roles:

- User: is able to use tools, create and use his own toolchains and use global toolchains;
- Administrator: on top of the user role, it is able to manage users, manager tools and manage global toolchains.

4.4.3.1. Deleting a user

An administrator can delete a user by clicking on the red cross at the top-right corner of the box. A confirmation message pops up, giving the possibility to cancel or confirm the deletion. When deleted properly, a success message is displayed at the top of the page. Otherwise, a red message is displayed, informing the user that a problem occurred during the deletion.

4.4.3.2. Adding a role

The administrator can add a role for a user, using the combobox listing all possible roles. He/she has to select a role (among User, Administrator, Supervisor) and to click on the "Add" button.

4.4.3.3. Deleting a role

The administrator can also change the role of a user through the following two methods:

- one by one, using the "Delete" button in front of the role to delete;
- all in one, using the "Delete all roles" button located in the "Manage Roles" section.

4.4.4. Manage tools and applications

On this page, administrators are able to add, delete and modify tools. We distinguish tools and applications. Tools can be internal or external thus can run:

- On a server managed by itrust consulting (those we call applications, only internal);
- On an external server and a link is provided to access it;
- On the client's browser and a link is provided to access it.

4.4.4.1. Manage tools.

In order to add a new tool, the administrator has to select the "Add" option from the “Manager” (tool management box) on the left hand side of the screen, in the **Tools** section, as shown in Figure 4.7.

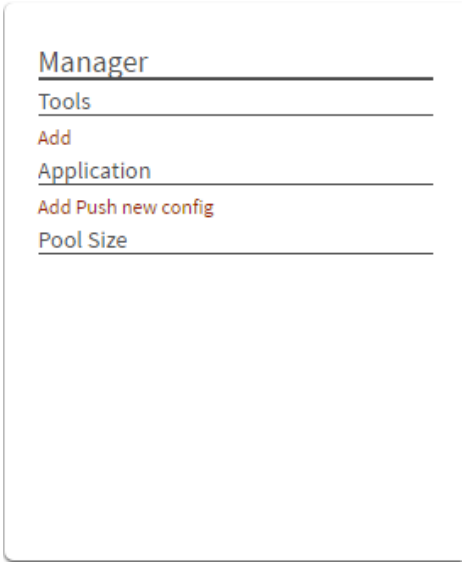

Figure 4.7.: Tool "Manager".

The tool edition page appears, as show in Figure 4.8. The administrator needs to complete all the following fields and save the changes.

- Name: name of the tool which is displayed on tool page;
- Publisher: contains the name of the publisher of the tool;
- Action name: describes how the application runs (internal or external tool);
- URL: the URL to use in order to launch the tool;
- Description: contains a description of the tool;
- Category: enables to choose one to several categories for the tool. The final categories are Visualisation, Data collection, Modelling, Analysis, and Misc;
- Type: the type of the tool;
- Application: enables to connects the tool to an application;
- Internal: defines whether the tool is internal to the interface or not;
- Run on application server
- Has input: defines if the tool requires input;
- Custom parameters: enables the user to set parameters for the application (in case of multiple options, the administrator delimits the parameters by semicolons, e.g. the value "Option1;Option2;Option3" will then appear in the form of a selection list which will include these three options);

- Links: enables the user to add input file sample, so any user can download it and test the tool.

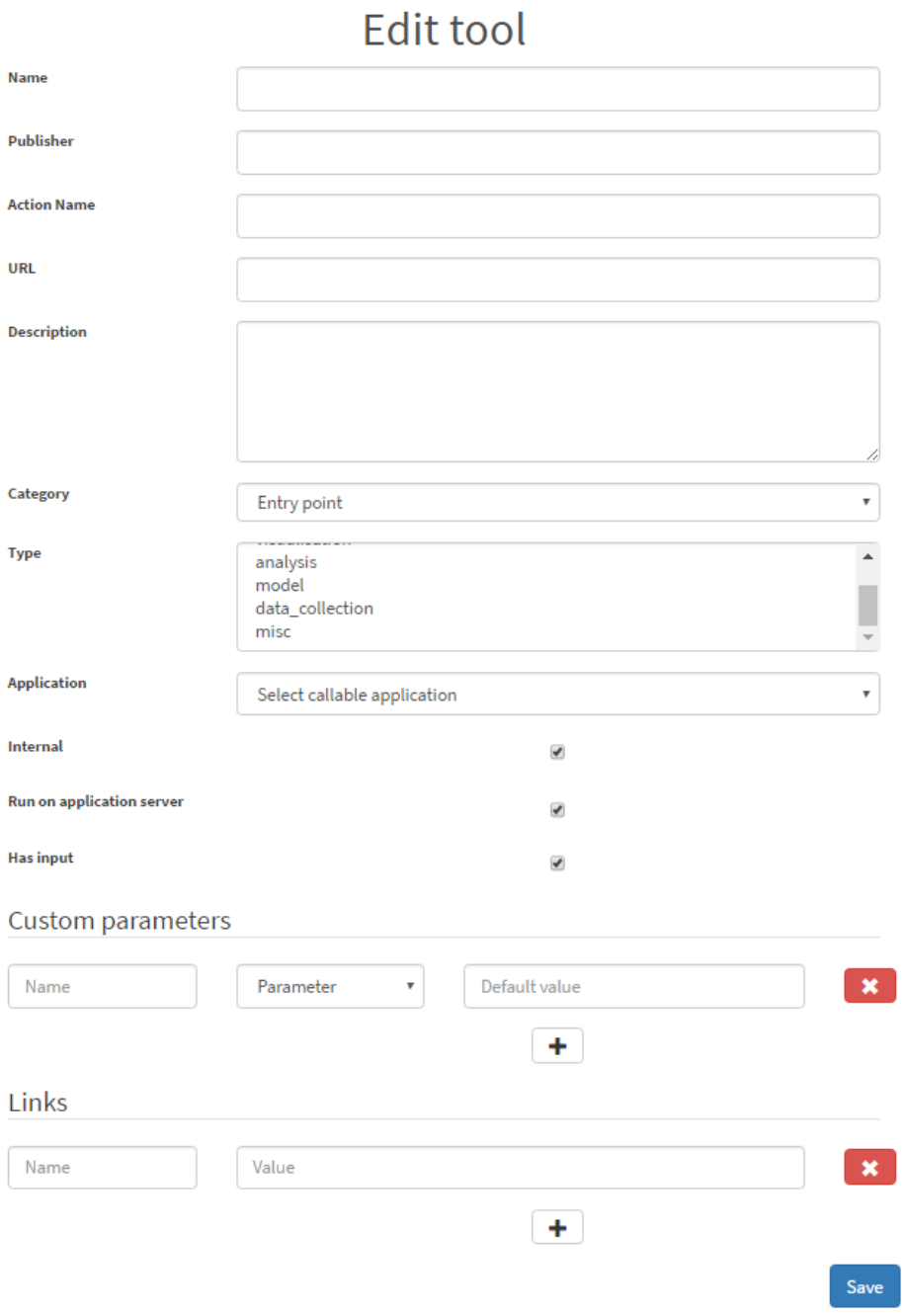

Figure 4.8.: Adding a new tool.

The following information is displayed (see Figure 4.9) for each tool in the TRESPASS platform:

- Path: the path to the executable to be run relative to the root of the tools
- Output file: the name of the main output file
- Required RAM size: the memory needs of the tool to limit the number of simultaneous applications (non-implemented as not needed)
- Required directory: Needs to be selected if the same application is run multiple times to avoid different instances of the same file to be mixed.
- Read from standard out: In case the output of a program is printed to the standard out buffer and not to a file.

Figure 4.9.: Tool information.

To delete a tool, the administrator has to click on the delete link (red cross at the top-right corner of the box.). A confirmation box opens, providing a button to cancel the deletion, and a button to confirm the deletion. When deleted properly, a success message is displayed at the top of the page. Otherwise, a red message is displayed, informing the user that a problem occurred during the deletion.

4.4.4.2. Manage applications.

In order to add an application, the administrator needs to click on the "Add" link in the **Application** section. It opens the form shown in Figure 4.10, which requires following information:

- Name: contains the name for the application;

- Path: path of the executable application in the software container;
- Output file: path of the output file after the execution of the application;
- Required RAM size: defines a RAM requirement;
- Required directory: defines whether the application requires a temporary directory or not;
- Read from standard out: defines whether the application reads input from standard in (ASK EOM WHY standard out) or not;
- Arguments: enables to add a list of arguments for the application.

Figure 4.10.: Adding an application.

To edit an application, click on edit icon.

It opens the form shown in Figure 4.12 with inputs already described above.

After adding or editing an application, the administrator has to push the new configuration. To do so, he/she must click on the "Push new config" link in the **Application** section, enter his/her administrator credentials in the authentication server page (see Figure 4.11), and click on the "Push" button.

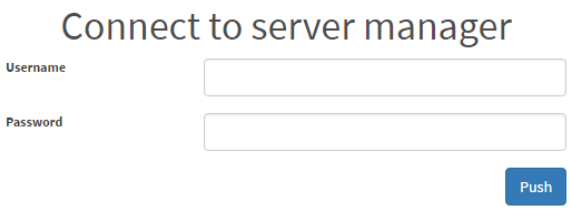

Figure 4.11.: Push the new configuration of an application.

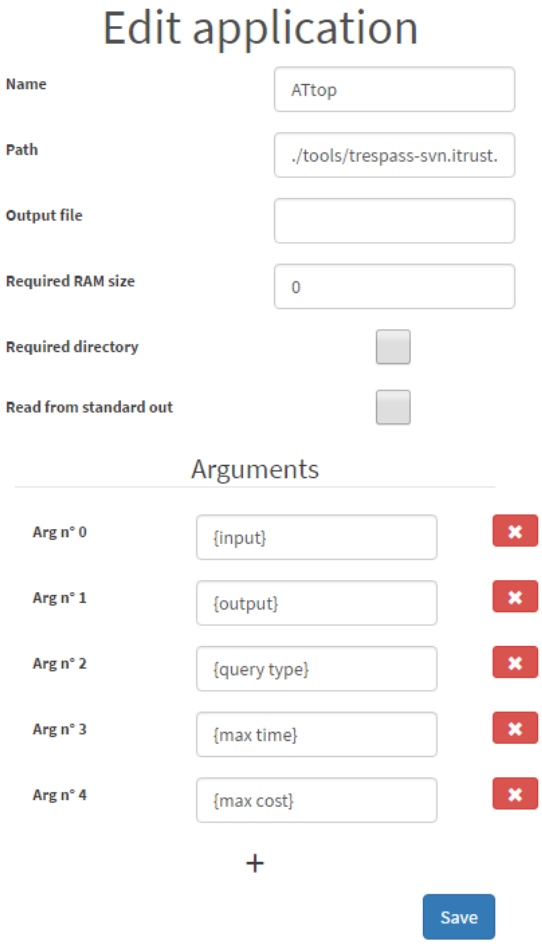

Figure 4.12.: Editing an application.

To delete an application, click on delete link linke for the tools.

4.4.5. Tools menu

The tools page shows the different tools that have been integrated. The possible actions for each tool are (three buttons on the top right corner of each box; only one button for default user):

- **Edit** to modify tool information (only for administrator).
- **Run** to execute the tool.
- **Delete** to remove the tool (only for administrator).

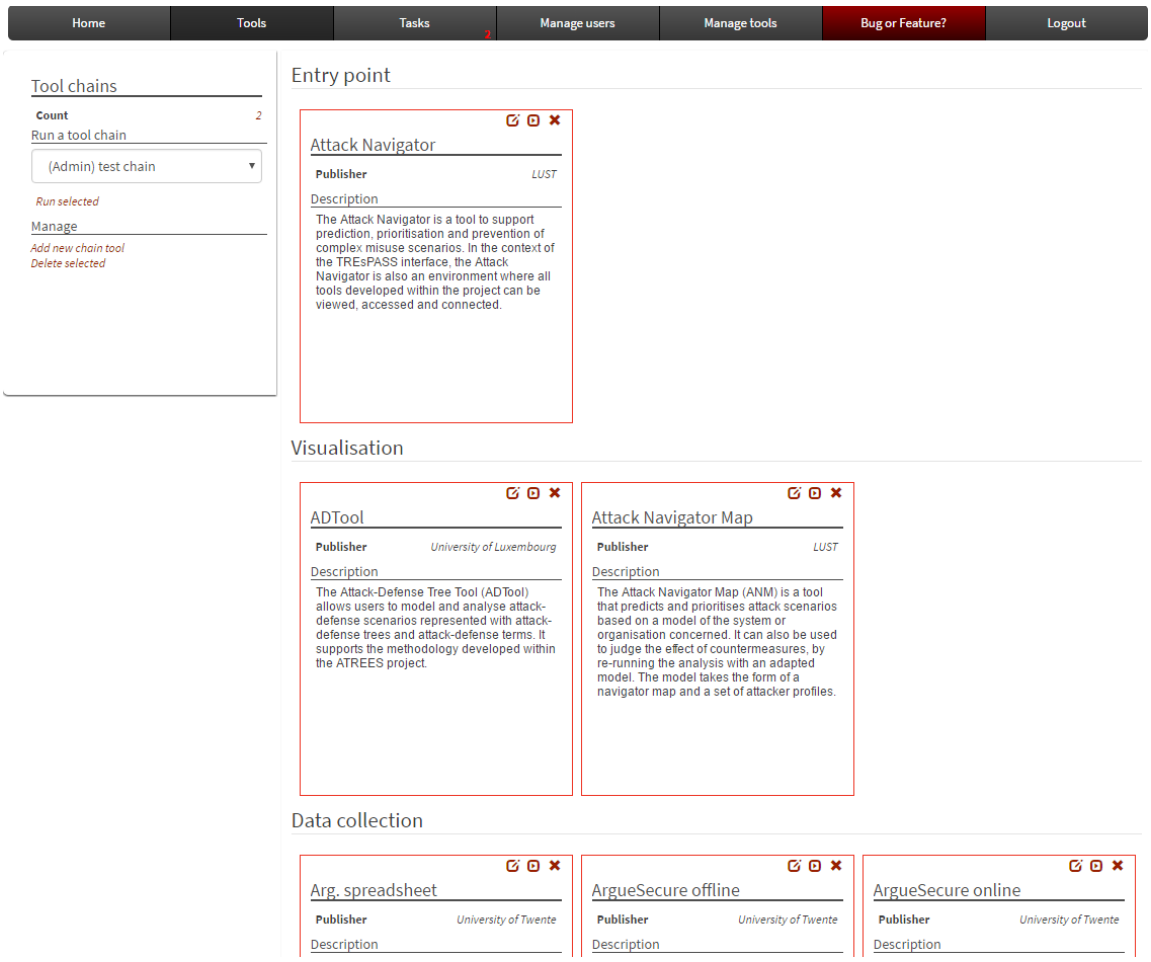

Figure 4.13.: Tools page.

4.4.6. Tasks menu

This page shows a list of tasks the user launched. Each summary contains following information:

- The **name** of the tool launched;
- The **status**, which shows whether the task finished successfully or with error;
- **Started at** / **Finished at** to see time information;
- **Processing rate** which depicts the progression of the task;
- **Input data** and **Output data** of the task, which is downloadable by the user.

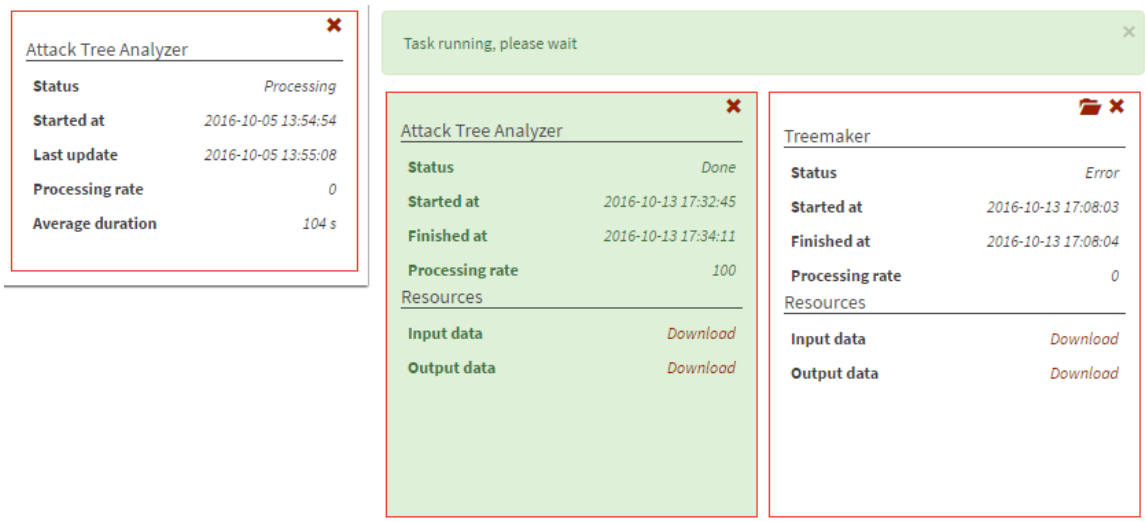

Figure 4.14.: Task menu.

To delete a task, the user has to use the icon located at each top-right corner of summary boxes. A confirmation box opens, providing a button to cancel the deletion, and another one to confirm the deletion. When deleted properly, a success message is displayed at the top of the page. Otherwise, a red message is displayed, informing the user that a problem occurred during the deletion.

4.4.7. Logout

When users want to exit the TRE$_S$PASS platform, they only need to click on the "Logout" link in the menu bar. They will be redirected to the initial authentication page, already shown in Figure 4.1.

4.5. Additional supporting features

In this section we list the features that are currently in place to assist the development and the use of TRE$_S$PASS tools.

4.5.1. Bug and feature tracker

In major collaborative software development projects it is worth having a set of tools to request features and report bugs. The integration team is familiar with the open source Redmine tool (www.redmine.org) and has configured an instance of this tool dedicated to TRESPASS. Individual users have to create an account and are invited to use it for the communication and tracking of technical aspects related to the prototype. The tool is available at http://trespass-ticket.itrust.lu.

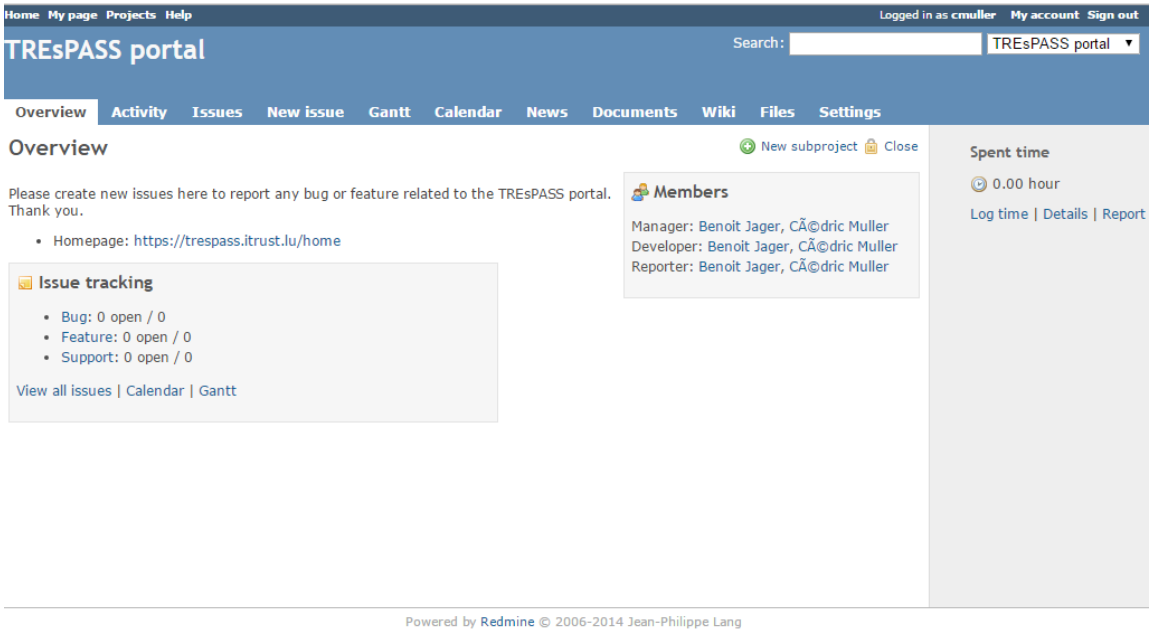

Figure 4.15.: Bug and feature tracker page.

A link to access the bug and feature tracker is available in the main menu bar. It allows to report any registration, login, technical or functional issues related to the TRESPASS platform. Once on the bug and feature tracker page, users need to click on the "TRESPASS portal" link or in the "Projects" menu at the top of the screen, as shown in Figure 4.15, then on the "New issue" link.

Users will have to provide the type of issue, subject, description, priority, due date, and assign the issue to one of the administrators in the list, then click on the "Create" button, or "Create and continue" button in case there are several issues to report. The corresponding page is shown in Figure 4.16.

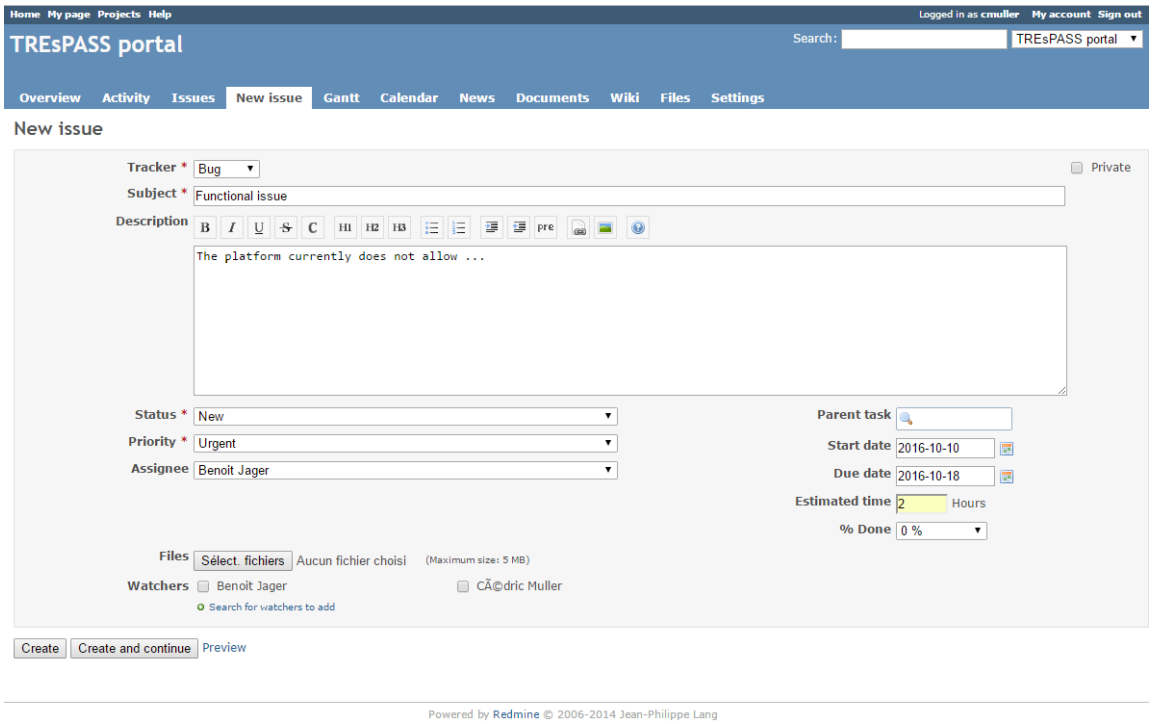

Figure 4.16.: Report a new issue in the bug and feature tracker.

It is possible to exit the bug and feature tracker after the issue has been reported, by clicking on the "Sign out" link in the top-right corner of the screen. Users will be informed that they successfully logged out, as shown in Figure 4.17.

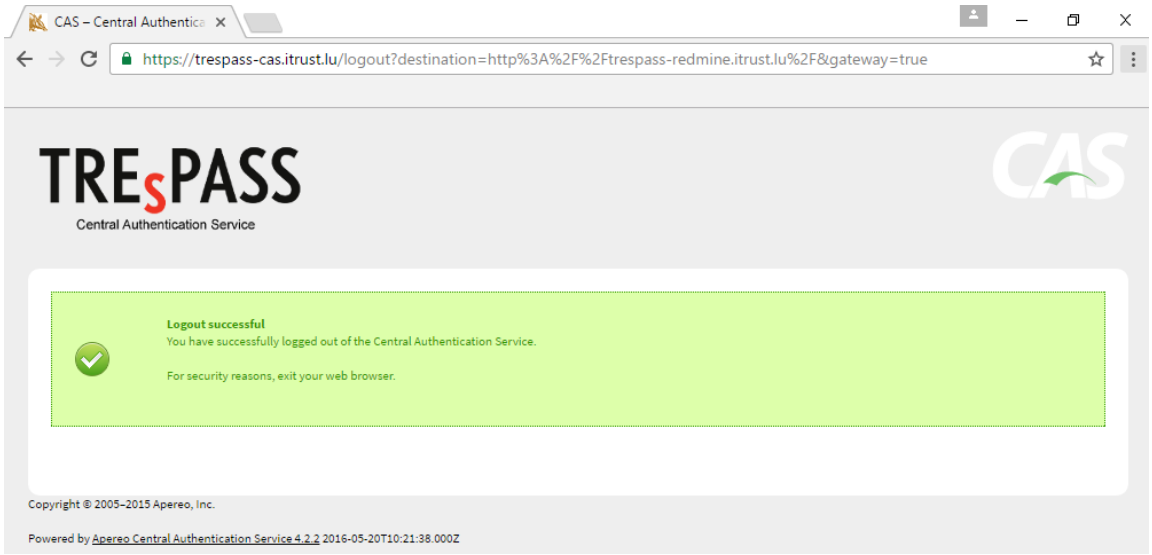

Figure 4.17.: Logout from the bug and feature tracker.

4.5.2. Software versioning

An SVN repository is available for the development of any individual tools. Git can also be included if needed. The reasons to provide this feature are to:

- Allow the collaborative development of TRESPASS tools.
- Allow partners to test their tools on the central platform without needing to ask the integration team to update a binary, dependencies, etc.
- Allow partners to keep their tools up to date, upgrade and maintain them, add new features, functionalities, etc.

The SVN repository is available at https://trespass-svn.itrust.lu. Most of the tools available on the website are on the SVN. Individual credentials can be requested from the integration team.

4.5.3. Common data and database models

In order to allow interoperability of the different components implemented in WP 1-5, common XML data (Figure 4.18) and database (Figure 4.19) models have been proposed. They can be found in the project SVN (trunk/WorkPackages/wp2/Tools/DataModel)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="trespass">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="accesspolicy" type="accesspolicyType">
                    <xs:key name="accesspolicy_PrimaryKey_1">
                        <xs:selector xpath="."/>
                        <xs:field xpath="ID"/>
                    </xs:key>
                    <xs:keyref name="accesspolicy_credential_ForeignKey
                    refer="accesspolicy_PrimaryKey_1">
                        <xs:selector xpath="accesspolicy_credential'
                        <xs:field xpath="IDAccessPolicy"/>
                    </xs:keyref>
                    <xs:keyref name="accesspolicy_data_ForeignKey_1"
                    refer="accesspolicy_PrimaryKey_1">
                        <xs:selector xpath="accesspolicy_data"/>
                        <xs:field xpath="IDAccessPolicy"/>
                    </xs:keyref>
                    <xs:keyref name="accesspolicy_identity_ForeignKey_1
                    refer="accesspolicy_PrimaryKey_1">
                        <xs:selector xpath="accesspolicy_identity"/>
```

Figure 4.18.: Common XML data model.

```
CREATE TABLE `accesspolicy` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(64) CHARACTER SET latin1 NOT NULL,
  `Notes` varchar(1024) CHARACTER SET latin1 DEFAULT NULL,
  `Active` bit(1) NOT NULL DEFAULT b'1',
  `Created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `CreatedBy` varchar(64) CHARACTER SET latin1 NOT NULL DEFAULT 'TRESPASS
  `Updated` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `UpdatedBy` varchar(64) CHARACTER SET latin1 NOT NULL DEFAULT 'TRESPASS
  PRIMARY KEY (`ID`),
  KEY `AccessPolicyIDX1` (`Name`)
);

CREATE TABLE `action` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(64) NOT NULL,
  `Notes` varchar(1024) DEFAULT NULL,
  `Active` bit(1) NOT NULL DEFAULT b'1',
  `Created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `CreatedBy` varchar(64) CHARACTER SET latin1 NOT NULL DEFAULT 'TRESPASS
  `Updated` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `UpdatedBy` varchar(64) CHARACTER SET latin1 NOT NULL DEFAULT 'TRESPASS
  PRIMARY KEY (`ID`),
  KEY `ActionIDX1` (`Name`)
```

Figure 4.19.: Common database model.

4.5.4. Security

The security of the TRE$_S$PASS tools has been considered during the design and the implementation of the prototype. The following security elements are included in the TRE$_S$PASS platform:

- **Access control:** TRE$_S$PASS tools are only available to authorised users. Users are required to create an account and log in with individual credentials before being able to use any tools. Minimum username and password lengths have been defined to four and eight characters, respectively. The access control has been implemented to avoid that people external to the project make use of the tools.

  Four administrators receive an email when an account is created and have to explicitly check if the email is from a member of the project and, if yes, validate the account.
- **CAPTCHA:** In order to prevent attacks from robots, a CAPTCHA system (Completely Automated Public Turing test to tell Computers and Humans Apart) has been used in the registration page.
- **Encryption:** All communications between the user's computer and the front-end TRE$_S$PASS server are encrypted using HTTPS (SSL/TLS) technology (Figure 4.20). This prevents the disclosure of access credentials when the server is accessed through insecure networks.

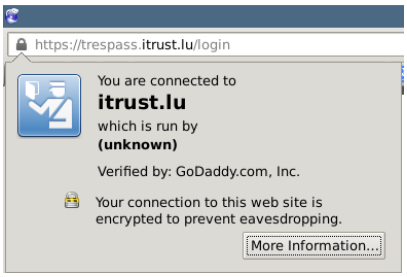

Figure 4.20.: Encrypted HTTPS connection.

4.5.5. API

Other tools or user interfaces, such as the Attack Navigator, may use the Applications Programming Interface (API), in order to access the programs in the SVN. In such cases, additional information about the API is available at the following link.

https://trespass.itrust.lu/api/json

5. Integration guide

5.1. Integration

New tools can be easily integrated. The procedure is slightly different depending on the platform that executes the tool:

- On a server managed by itrust: the application has to be installed before adding the tool;
- On an external server: a link to the external server is provided;
- On the client's browser: the tool is provided. Often the tool is written in Java and the jar file is provided.

We can define the Java Network Launching Protocol (JNLP) file with signature so the client's browser can recognise the tool as trustworthy. Visualisation tools are mainly JavaScript web apps that also run in the browser, but these do not need signature (see section 7.4.2 for more details).

5.2. The integration level

Reference Model for Frameworks of Software Engineering Environments (NIST/ECMA TR55) identifies five integration dimensions:

- data: ability to share information throughout the environment using common database and data formats.
- control: ability to combine the functionalities offered in an environment in flexible ways.
- presentation: ability to interact with the environment functionalities with similar screen appearance and similar modes of interaction.
- process: ability to access environment functionalities based upon a pre-defined enactable development process.
- framework: degree to which tools are integrated with the framework, the provision of services for common mechanisms for installation, modification and deletion of tools.

On the data dimension, common database and data formats were defined but they are not used by all tools as the priority is still on the main features of each tool. On the control dimension, there is a high degree of flexibility in line with requirements by partners and most of them are either already in place or under final review. On the presentation dimension, there has been a effort to reuse most of the interfaces for the different tools, providing the user with a seamless experience. On the process dimension there is a bigger deviation justified by the fact that the project focus on innovative socio-technical security features rather than pure software development. Finally, on the framework dimension a high level of integration is available as adding or modifying tools can be performed without intervention from the administrators through the developed interfaces or the supporting features.

5.3. Acceptance testing

Functional tests are executed by the administrators and by the tool developers to check that the tool is working as expected.

5.4. Security testing

Regular security tests are applied to the TRE$_S$PASS platform.

5.5. Maintenance

The maintenance of the platform consists of keeping up-to-date software packages, applying bug fixes to the platform and individual tools, and the execution of functional tests after maintenance operations. The update and upgrade of the operating system is achieved simply by running standard commands:

```
# apt-get update && apt-get upgrade
```

See sections 6.3.2 and 7 to get more details about the update of services and the commands to be executed for the deployment and the maintenance of dockers.

5.6. Requirements for tool integration

In order to address the definition of requirements in a systematic way, a project-wide task-force was put in place. In that context, all technical work packages were requested to identify their most relevant requirements. For a full overview of the requirements identified by the task-force, we refer to (The TRESPASS Project, D6.2.2, 2015).

The list of the requirements that are targeted at WP6, and the discussion about their implications are available in more details in Appendix B. An update about the final implementation status has been provided.

5.7. Integrated tools in more details

The TRESPASS consortium has decided for the adoption of a loosely coupled solution as justified in (The TRESPASS Project, D6.1.2, 2015), using a centralised integration component, which we call the TRESPASS Core System (TCS). In this way, several modules coming from the different work packages of the project should co-operate through this core system.

5.7.1. WP1 Modelling tools

5.7.1.1. Architect, BiZZdesign

Architect is a leading software tool for Enterprise Architecture. Architect is compliant with ArchiMate and TOGAF, the open standards for Enterprise Architecture, maintained by The Open Group. Architect is used for creating, editing, visualising and analysing ArchiMate models (including extensions for modelling risk and security aspects). With a specific configuration developed in the project, also TRESPASS models (infrastructures and processes) can be created and edited.

The access to Architect is through RDP and dedicated credentials are needed.

More information: http://bizzdesign.com/tools/bizzdesign-architect/

5.7.2. WP1 Attack generation tools

5.7.2.1. TRESPASS Model, Technical University of Denmark

As mentioned in (The TRESPASS Project, D1.3.1, 2013), the first prototype of the TRESPASS model implements part of the final functionality of the model. It is a proof of concept of the techniques that will establish the core of the TRESPASS tools, storing the data and making it available to analyses, visualisation, and to the TRESPASS process.

TRESPASS model tool generates an attack tree in the XML format defined by University of Luxembourg within ADTool, which is a common basis input format for other tools.

5.7.2.2. Treemaker, Technical University of Denmark

This tool generates an attack tree in the XML format that can be used for analyses, visualisation, and for the TRESPASS process. It takes as input the description of the model in XML format. Additional information about Treemaker is also available in deliverable D3.4.2 (The TRESPASS Project, D3.4.2, 2016).

5.7.3. WP2 Data Extraction tools

These tools, most of them existing tools, will be integrated at a later stage in the prototype. They are listed in (The TRESPASS Project, D2.2.1, 2013). Their aim is to extract data from a digital infrastructure that could be of value as input to the risk modelling and visualisation processes. Extracted data provide information about systems, networks, software and configuration.

5.7.4. WP3 Quantitative analysis tools

Quantitative analysis tools have been developed and adopted within the project in order to accommodate TRESPASS requirements.

5.7.4.1. ADTool, University of Luxembourg

The tool was developed before TRESPASS but has been adapted to meet the project requirements. Version 1.2 was issued at the beginning of October 2013 and includes the following main improvements with respect to the previous version:

- possibility of using multiline labels;
- support for exporting and importing trees written in the XML format de defined in (The TRESPASS Project, D1.3.1, 2013). An example of the IPTV attack tree in this XML format refer to (The TRESPASS Project, D3.3.1, 2013).

ADTool is used for the construction and manipulation of the first attack tree. The tool has its own GUI (Graphical User Interface) to manipulate attack trees. Currently the tool can be downloaded from the TRESPASS platform and needs to be executed in the user's computer. A deeper integration level is possible by entirely reworking the GUI and converting it into a web-based technology, which appears to bring no significant advantages. Other tools that require the input of ADTool XML-based output files will have an HTML form on the website to upload files.

More relevant information about the tool can be found in the deliverable D3.4.2 (The TRESPASS Project, D3.4.2, 2016) and http://satoss.uni.lu/members/piotr/adtool/.

5.7.4.2. Attack Tree Analyser (ATA), Cybernetica

The initial version of this tool was developed before TRESPASS and has been adapted to the file formats used in the project. The attack tree computation tool called Attack Tree Analyser (ATA) can be used to calculate optimal attack vectors (from the attacker point of view). The fact of taking the attacker profiles into account is also an improvement that occurred in the scope of TRESPASS. Additional information about the ATA is also available in deliverable D3.4.2 (The TRESPASS Project, D3.4.2, 2016).

Failure-free Model was a tool used to assess if a system is secure against rational gain-oriented attackers. It took an attack tree (with parameters for atomic actions) as input and it is now fully integrated into the ATA tool as one of its operating modes. Converter was also a previous developed stand-alone tool to convert the output format of the TRESPASS model (XML) into the input format understandable by the ATA tool and it is now fully integrated too.

5.7.4.3. TRICK Service, itrust consulting

TRICK Service (Tool for Risk management of an ISMS based on a Central Knowledge base) is a risk assessment and management tool for identification, analysis and estimation of assets, threats, vulnerabilities, risk scenarios and security measures. It can be used to conduct risk assessments according to ISO/IEC 27005 or CSSF 12/544.

It enables to determine a list of security measures to implement in order to reduce the impact or the occurrence likelihood of possible risk scenarios. It assesses quantitative risks and fulfils risk treatment operations according to the user-configured referential standards (ISO/IEC 27001, 27002, etc.).

It outputs the risk treatment plan which prioritises the implementation of security measures according to their Return On Security Investment (ROSI), risk specificities and feasibility.

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 TRESPASS project.

An approach to automated selection of preventive security controls has also been developed in the context of the TRESPASS project. It is supported by a tool called ADTop, which can help organisations to fine-tune their risk assessments and build their own libraries of attacks and countermeasures in the real-world situation. ADTop actually extends the TRICK Service, and has been validated thanks to the realistic case study called ÉpStan documented in D7.4.2 (The TRESPASS Project, D7.4.2, 2016).

A new module for dynamic risk analyses has also been developed and integrated in TRICK Service. The assessment made by itrust consulting for dealing with new security situations

in risk assessment is described in more details in the deliverable D5.3.3 (The TRESPASS Project, D5.3.3, 2016).

TRICK Service and ADTop are described in more details in deliverables D3.4.2 (The TRESPASS Project, D3.4.2, 2016) and D5.4.2 (The TRESPASS Project, D5.4.2, 2016) and their usage are demonstrated throughout various use cases in deliverables D7.2.2 (The TRESPASS Project, D7.2.2, 2016) and D7.4.2.

5.7.4.4. ATCalc, University of Twente

This tool was developed before TRESPASS and a conversion script is needed in order to adapt the formats.

ATCalc is a tool for efficient attack tree analysis. It allows time dynamic analysis of attack trees. It computes the system unreliability for each mission time, i.e. the probability that the system fails within the mission time. Furthermore it is capable of computing the Mean Time To Failure (MTTF), i.e. the expected time that the system will fail.

ATCalc can be used by downloading a stand-alone version accessible via git from `http://fmt.cs.utwente.nl/tools/scm/dftcalc.git` in the branch atcalc, and via a web interface accessible at `http://fmt.ewi.utwente.nl/puptol/atcalc/` shown in Figure 5.1.

ATCalc is open source, but requires a license for CADP, which is free for academic institutions. ATCalc allows user to:

- Input AT(Attack tree) models via a text screen;
- Run analysis over several dependability metrics. This can be:
  - The probability of an attack for several attack times t, or
  - The probability on an attack during an interval [T1; T2], or
  - The mean time for a successful attack.

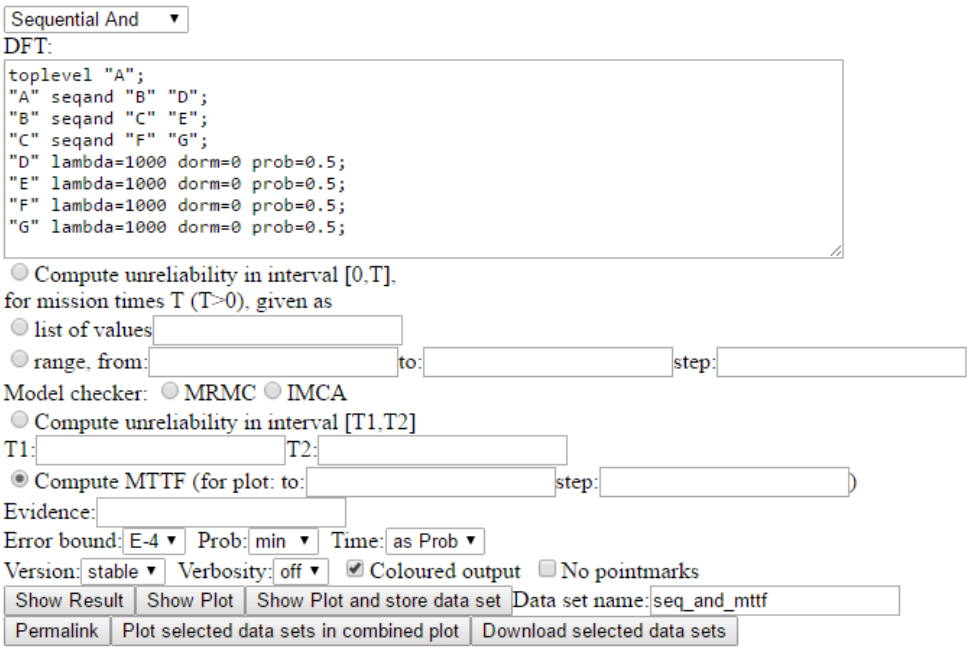

Figure 5.1.: ATCalc.

The deliverable D3.4.2 (The TRESPASS Project, D3.4.2, 2016) contains all the updated information on ATCalc and ATtop.

5.7.5. WP4 Visualisation tools

The conceptual tools include the visualisation atlas developed for the first WP4 (The TRESPASS Project, D4.1.1, 2013) deliverable and the initial expressivity measurement tool developed as part of the same deliverable. These conceptual tools are available on https://trespass.itrust.lu/tools.

5.7.6. WP5 Processes

5.7.6.1. Attack Pattern Library (APL), Cybernetica

To extend the basic attack trees exported by the tools of WP1 with more detailed attack descriptions and parameter evaluations, an extension library has been developed. This library called the Attack Pattern Library (APL) is further specified in Deliverable 5.3.1 (The TRESPASS Project, D5.3.1, 2013).

Attack pattrern library has been integrated into the TRESPASS Knowledge Base (TKB) developed by IBM. Its main technical component is a distributed revision control system (like Git), which allows implementation of different information sharing use cases. The first data items (essentially, small attack trees) in the library can be obtained from the existing

libraries (CAPEC, ISKE, SVRS) and hand-tuned by the TRESPASS consortium experts. As soon as the first data items are available, the setup of the above-mentioned control system is straight-forward.

For the description of its internal representation as well as the details on integration we refer the reader to Deliverable 2.4.1 (The TRESPASS Project, D2.4.1, 2016).

6. Deployment guide

6.1. Description of deployment package

The following sections describe how to deploy an instance of the TRE$_S$PASS platform.

6.2. Pre-deployment checks

6.2.1. Requirements for deployment

The prototype should be deployed on any operating system packaging:

- Docker v1.11.2 and newest;
- Docker Compose v1.5.2 and newest.

Docker is a technology which provides automation for the deployment of Linux applications inside software containers. A software container is a virtualisation method where the OS kernel authorises multiple isolated user-space instances. It can be seen as an advanced implementation of the chroot mechanism.

6.2.2. Configuration

The configuration of the containers is described in more details in section 6.3.2.

6.3. Deployment procedure

6.3.1. Description of components

6.3.1.1. LDAP

This container runs a LDAP server, which contains the information on the user registered to the service platform (e.g. firstname, lastname, password hash).

6.3.1.2. CAS

This container runs CAS, a single-sign on solution provided by Apereo, and implementing the CAS protocol. This provide a single-sign on solution to all public container of the prototype except for the SVN container, which must stay compatible with SVN clients. This container is connected to the ldap container, as it has to get user information to authenticate users.

6.3.1.3. SVN

This container runs the SVN repository where users can commit their software. It is connected to the ldap container to authenticate users.

6.3.1.4. MySQL

The mysql container runs an instance of MySQL in order to store database for the Front-End, Redmine, Software Checker, ArgueSecure, and TRICK Service services.

6.3.1.5. Software

The software container runs a service which listens and launch tools when someone runs a task on the Front-End service. It needs a connection with the sc container in order to communicate with Software Checker.

6.3.1.6. FE

This container runs the Front-End (FE) service of the TRE$_S$PASS platform, which provides features to add and run tools and chains. It is connected to the mysql container to store data, and to the software container to run different internal tools.

6.3.1.7. SC

The sc container runs an instance of Software Checker tool. It needs a connection with mysql to store its data.

6.3.1.8. TKB

This container is a cherrypy instance running tkb.

6.3.1.9. TKB-FE

This container is a front-end for the tkb container which enables authentication against the cas container.

6.3.1.10. Redmine

The redmine container runs the Redmine ticketing system. It needs a connection with the mysql container to store its data, and with the cas container to authenticate users.

6.3.1.11. Trickservice

The trickservice container runs a TRICK Service instance. It needs a connection with the mysql container to store its data, and with the ldap container to authenticate users.

6.3.1.12. Interactor-db

The interactor-db container is a Mongo database used to store data for the interactor tool.

6.3.1.13. Interactor

This container runs an interactor instance. It needs a connection with the interactor-db container.

6.3.1.14. Arguesecure-node

This container runs a node js server used by ArgueSecure tool.

6.3.1.15. Arguesecure-redis

This container runs a redis server used by ArgueSecure tool.

6.3.1.16. Arguesecure-memcached

This container runs a memcached server used by ArgueSecure tool.

6.3.1.17. Arguesecure

This container runs an ArgueSecure online instance.

6.3.1.18. Network communication

The following figure shows the network communications that are needed and permitted between containers running the prototype. A green box depicts a container directly reachable by users of the prototype. Direction of arrow depicts the dependency between containers: in our case, **fe** needs to communicate with **cas** in order to validate session tickets that it receives from users.

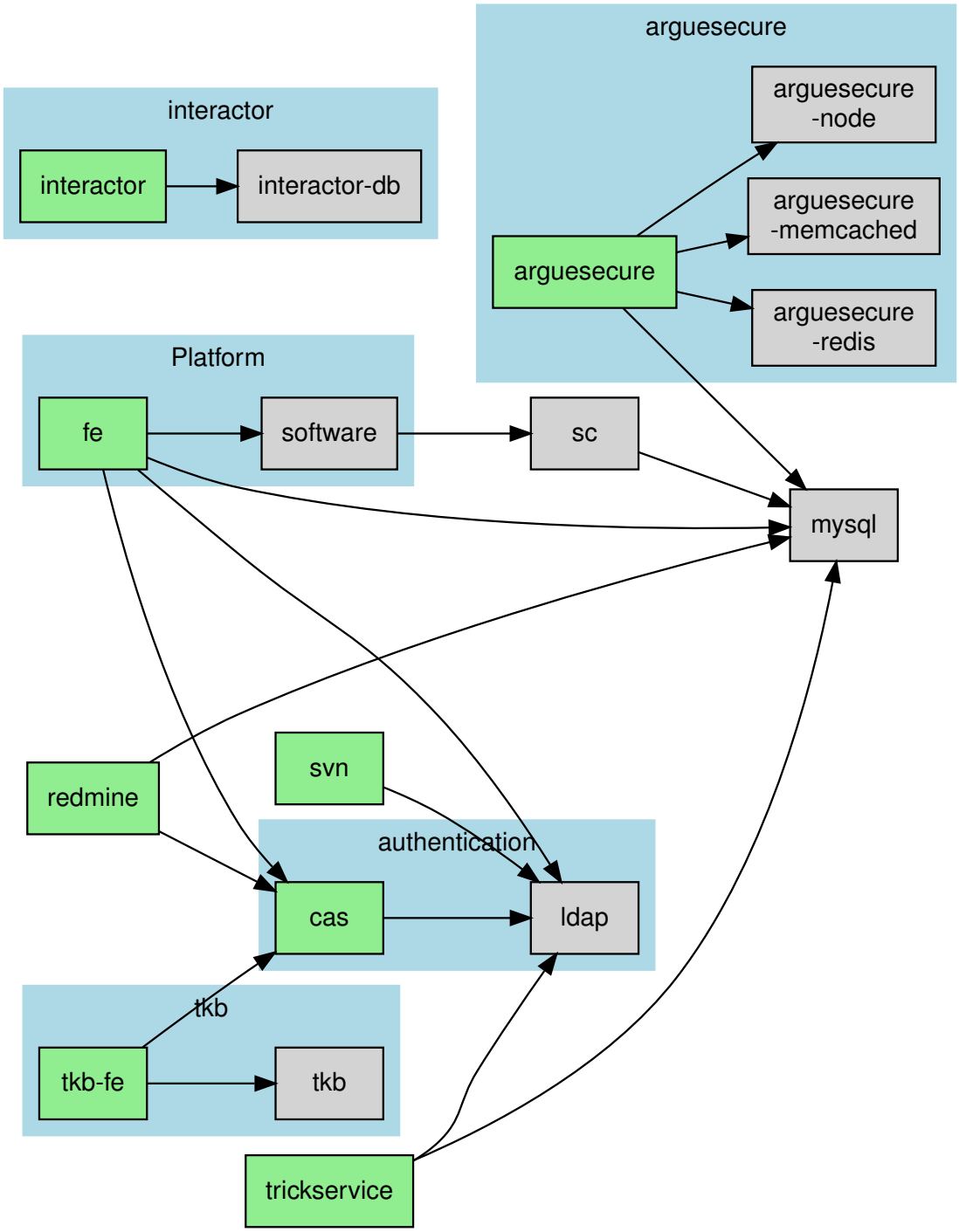

Figure 6.1.: Network communication between containers.

6.3.2. Deployment of the prototype

6.3.2.1. Download the docker files

These steps clone the docker repository of the trespass project from GitHub. They also checkout the local repository to the master branch, in order to get the stable release.

```
git clone https://github.com/trespass/docker.git trespass-docker
cd trespass-docker
git checkout master
```

6.3.2.2. Prerequisite

The shell commands given in next sections requires shell being located in the **trespass-docker** folder created during previous step.

6.3.2.3. Build the images

This step builds images of all components of the prototype. The build takes several minutes to finish, depending on the network bandwidth and processor capacity.

First, copy environment example files as environment files. It is required as build command will check for their existence:

```
cp env/examples/* .
```

Then, launch the build:

```
docker-compose build
```

6.3.2.4. Configure and start the LDAP container

Edit **ldap.env** file and configure following environment variables:

- **SLAPD_DOMAIN**: base DC (Domain component) of LDAP
- **SLAPD_PASSWORD**: password for admin account
- **SLAPD_CONFIG_PASSWORD**: password to modify configuration without restarting LDAP.

Following is an example of configuration:

```
SLAPD_DOMAIN=ldap.itrust.lu
SLAPD_PASSWORD=p@ssW0rd
SLAPD_CONFIG_PASSWORD=p@ssW0rd
SLAPD_ADDITIONAL_SCHEMAS=ppolicy
```

Then, start the container:

```
docker-compose up -d ldap
```

At this point, you can see logs of container with following command:

```
docker-compose logs ldap
```

6.3.2.5. Configure and start the MySQL container

Edit **mysql.env** file and configure following environment variables:

- **FE_DB_NAME**: name of database for trespass front-end
- **FE_DB_USER**: database user for trespass front-end
- **FE_DB_PASS**: password of database access for trespass front-end
- **SC_DB_NAME**: name of database for software checker
- **SC_DB_USER**: database user for software checker
- **SC_DB_PASS**: password of database access for software checker
- **REDMINE_DB_NAME**: name of database for Redmine
- **REDMINE_DB_USER**: database user for Redmine
- **REDMINE_DB_PASS**: password of database access for Redmine
- **ARGUESECURE_DB_NAME**: name of database for ArgueSecure
- **ARGUESECURE_DB_USER**: database user for ArgueSecure
- **ARGUESECURE_DB_PASS**: password of database access for ArgueSecure
- **TRICKSERVICE_DB_NAME**: name of database for TRICKService
- **TRICKSERVICE_DB_USER**: database user for TRICKService
- **TRICKSERVICE_DB_PASS**: password of database access for TRICKService

Following is an example of configuration:

```
FE_DB_NAME=trespass
FE_DB_USER=trespass
FE_DB_PASS=p@ssW0rd
REDMINE_DB_NAME=redmine
REDMINE_DB_USER=redmine
REDMINE_DB_PASS=p@ssW0rd
SC_DB_NAME=sc
SC_DB_USER=sc
SC_DB_PASS=p@ssW0rd
```

These environment variables will be used to create corresponding databases and user accesses. Then, start the container:

```
docker-compose up -d mysql
```

At this point, you can see logs of container with following command:

```
docker-compose logs mysql
```

Now, get the name of the container in **NAME_MYSQL_CONTAINER** environment variable, and create databases and user accesses as mentioned previously:

```
NAME_MYSQL_CONTAINER=$(docker-compose ps | grep "_mysql_" | awk '{print $1}')
docker exec -it $NAME_MYSQL_CONTAINER /init-databases-users.sql
```

6.3.2.6. Configure and start the SVN container

Edit **svn.env** file and configure following environment variables:

- **LDAP_ADMIN_USER**: CN of admin. The CN must respect domain defined with **SLAPD_DOMAIN** in 6.3.2.4. For instance, if **SLAPD_DOMAIN** is ldap.itrust.lu, then the CN will be cn=admin,dc=ldap,dc=itrust,dc=lu;
- **LDAP_ADMIN_PASSWORD**: password for admin account. It must be the same value as **SLAPD_PASSWORD** defined in 6.3.2.4.
- **LDAP_URL**: URL to LDAP server. It must contains the base as defined in 6.3.2.4.

cn=admin part of **LDAP_ADMIN_USER**, and **ou=people** part of **LDAP_URL** cannot be changed.

Following is an example of configuration:

```
LDAP_ADMIN_USER=cn=admin,dc=ldap,dc=itrust,dc=lu
LDAP_ADMIN_PASSWORD=p@ssW0rd
LDAP_URL=ldap://ldap:389/ou=people,dc=ldap,dc=itrust,dc=lu?cn?sub?(objectClass=*)
```

Then, start the container:

```
docker-compose up -d svn
```

At this point, you can see logs of container with following command:

```
docker-compose logs svn
```

6.3.2.7. Configure and start the CAS container

Edit **cas.env** file and configure following environment variables:

- **LDAP_URL**: does not need to be changed as **ldap** dns will work as **LDAP** container and **CAS** are linked through Docker. However, if you want to deploy **LDAP** and **CAS** on two separate servers, you have to change **ldap** with a public DNS.
- **SERVICE_JSON_SERVICEID**:
- **LDAP_ROOTDN**
- **LDAP_BASEDN**
- **LDAP_AUTHN_SEARCHFILTER**
- **CAS_LOGOUT_REDIRECT**
- **TOMCAT_NATIVE_LIBDIR**
- **LD_LIBRARY_PATH**
- **FE_URL**

Following is an example of configuration:

```
LDAP_URL=ldap://ldap:389
SERVICE_JSON_SERVICEID=^(http|https)://.*
LDAP_ROOTDN=dc=ldap,dc=itrust,dc=lu
LDAP_BASEDN=ou=people,dc=ldap,dc=itrust,dc=lu
LDAP_AUTHN_SEARCHFILTER=(&(cn={user})(!(userAccountControl:1.2.840.113556.1.4.803:=2))
(!(userAccountControl:1.2.840.113556.1.4.803:=8388608)))
CAS_LOGOUT_REDIRECT=true
TOMCAT_NATIVE_LIBDIR=/usr/local/tomcat/native-jni-lib
LD_LIBRARY_PATH=/usr/local/tomcat/native-jni-lib
FE_URL=https://tp.itrust.lu
```

Then, start the container:

```
docker-compose up -d cas
```

At this point, you can see logs of container with following command:

```
docker-compose logs cas
```

6.3.2.8. Configure and start the tkb and tkb-fe containers

Edit **tkb-fe.env** file and configure following environment variables:

- **CAS_LOGIN_URL**: public URL to login page of the CAS server. Change DNS with the public DNS of the CAS server;
- **CAS_ROOT_PROXIEDAS**: public URL to ANM server.
- **CAS_VALIDATE_URL**: does not need to be changed as **cas** dns will work as **tkb-fe** container and **CAS** are linked through Docker. However, if you want to deploy **tkb-fe** and **cas** on two separate physical servers, you have to change **cas** with a public DNS.

Following is an example of configuration:

```
APACHE_MODE=cherrypy
CAS_COOKIE_PATH=/var/cache/apache2/mod_auth_cas/
CAS_LOGIN_URL=https://tp-cas.itrust.lu/login
CAS_VALIDATE_URL=https://cas/serviceValidate
CAS_ROOT_PROXIEDAS=https://tp-tkb.itrust.lu
CAS_VERSION=2
AUTH_REQUIRE=valid-user
```

Then, start the container:

```
docker-compose up -d tkb tkb-fe
```

At this point, you can see logs of containers with following command:

```
docker-compose logs tkb tkb-fe
```

6.3.2.9. Configure and start the sc container

Edit **sc.env** file and configure following environment variables:

- **DB_HOST**: does not need to be changed as **mysql** dns will work as **mysql** container and **sc** are linked through Docker. However, if you want to deploy **mysql** and **sc** on two separate servers, you have to change **mysql** with the right IP or DNS.
- **DB_PORT**: IP port for MySQL server
- **DB_NAME**: database name on MySQL server. It must be the same value as **SC_DB_NAME** defined in 6.3.2.5.
- **DB_USER**: database user to authenticate to MySQL server. It must be the same value as **SC_DB_USER** defined in 6.3.2.5.

- **DB_PASS**: database password to authenticate to MySQL server. It must be the same value as **SC_DB_NAME** defined in 6.3.2.5.

Following is an example of configuration:

```
MYSQL_DATABASE=sc
MYSQL_USER=sc
MYSQL_PASSWORD=p@ssW0rd
MYSQL_PORT=3306
MYSQL_HOST=mysql
```

Then, start the container:

```
docker-compose up -d sc
```

At this point, you can see logs of container with following command:

```
docker-compose logs sc
```

6.3.2.10. Configure and start the fe container

Edit **fe.env** file and configure following environment variables:

- **RECAPTCHA_KEY**: key to activate the reCAPTCHA from Google. See ADD REFERENCE on how to get a reCAPTCHA key.
- **CAS_HOST**: public URL to the CAS server.
- **FE_HOST**: public DNS of the FE itself to be used for the links in emails sent to users.

Following is an example of configuration:

```
RECAPTCHA_KEY=
CAS_HOST=https://tp-cas.itrust.lu
FE_HOST=https://tp.itrust.lu
```

Then, start the container:

```
docker-compose up -d fe
```

At this point, you can see logs of container with following command:

```
docker-compose logs fe
```

6.3.2.11. Configure and start the redmine container

Edit **redmine.env** file and configure following environment variables:

- **DB_HOST**: does not need to be changed as **mysql** dns will work as **mysql** container and **redmine** are linked through Docker. However, if you want to deploy **mysql** and **redmine** on two separate servers, you have to change **mysql** with the right IP or DNS.
- **DB_PORT**: IP port for MySQL server
- **DB_NAME**: database name on MySQL server. It must be the same value as **REDMINE_DB_NAME** defined in 6.3.2.5.
- **DB_USER**: database user to authenticate to MySQL server. It must be the same value as **REDMINE_DB_USER** defined in 6.3.2.5.
- **DB_PASS**: database password to authenticate to MySQL server. It must be the same value as **REDMINE_DB_NAME** defined in 6.3.2.5.
- **CAS_URL**: public URL to the CAS server.

Following is an example of configuration:

```
DB_HOST=mysql
DB_PORT=3306
DB_NAME=redmine
DB_USER=redmine
DB_PASS=p@ssW0rd
CAS_URL=https://tp-cas.itrust.lu
```

Then, start the container:

```
docker-compose up -d redmine
```

At this point, you can see logs of container with following command:

```
docker-compose logs redmine
```

When Redmine has started completely (i.e. created database and migrating all plugins), get the name of the container in **NAME_REDMINE_CONTAINER** environment variable and configure plugin to connect Redmine with the CAS server:

```
NAME_REDMINE_CONTAINER=$(docker-compose ps | grep "_redmine_" | awk '{print $1}')
docker exec -it $NAME_REDMINE_CONTAINER /init-sql.sql
```

6.3.2.12. Configure and start the trickservice container

Edit **ts.env** file and configure following environment variables:

- **DB_HOST**: does not need to be changed as **mysql** dns will work as **mysql** container and **trickservice** are linked through Docker. However, if you want to deploy **mysql** and **trickservice** on two separate servers, you have to change **mysql** with the right IP or DNS.
- **DB_PORT**: IP port for MySQL server
- **DB_NAME**: database name on MySQL server. It must be the same value as **TS_DB_NAME** defined in 6.3.2.5.
- **DB_USER**: database user to authenticate to MySQL server. It must be the same value as **TS_DB_USER** defined in 6.3.2.5.
- **DB_PASS**: database password to authenticate to MySQL server. It must be the same value as **TS_DB_NAME** defined in 6.3.2.5.

Following is an example of configuration:

```
DB_HOST=mysql
DB_PORT=3306
DB_NAME=ts
DB_USER=ts
DB_PASS=p@ssW0rd
```

Then, start the container:

```
docker-compose up -d trickservice
```

At this point, you can see logs of container with following command:

```
docker-compose logs trickservice
```

6.3.2.13. Configure and start the arguesecure container

Edit **arguesecure.env** file and configure following environment variables:

- **DB_HOST**: does not need to be changed as **mysql** dns will work as **mysql** container and **arguesecure** are linked through Docker. However, if you want to deploy **mysql** and **arguesecure** on two separate servers, you have to change **mysql** with the right IP or DNS.
- **DB_PORT**: IP port for MySQL server
- **DB_NAME**: database name on MySQL server. It must be the same value as **ARGUESECURE_DB_NAME** defined in 6.3.2.5.

- **DB_USER**: database user to authenticate to MySQL server. It must be the same value as **ARGUESECURE_DB_USER** defined in 6.3.2.5.
- **DB_PASS**: database password to authenticate to MySQL server. It must be the same value as **ARGUESECURE_DB_NAME** defined in 6.3.2.5.

Following is an example of configuration:

```
DB_HOST=mysql
DB_PORT=3306
DB_NAME=arguesecure
DB_USER=arguesecure
DB_PASS=p@ssW0rd
```

Then, start the container:

```
docker-compose up -d arguesecure
```

At this point, you can see logs of container with following command:

```
docker-compose logs arguesecure
```

6.4. Recovery procedures in case of deployment failure

In case of failure in deployment of dockers (software), a downgrade will be proceed in order to return to previous stable version. In case of failure of the Operating System (OS), the administrator has to reinstall the OS from the beginning.

7. Maintenance guide

7.1. Preventive maintenance

7.1.1. Backup of data

We recommend making regular backups of the folder containing data used by containers. We recommend utilization of the Borg Backup solution. It is a deduplicating backup program featuring encryption and compression.

7.1.1.1. Borg backup solution

Create a repository The creation of a borg repository is made through the following command:

```
borg init /path/to/repo
```

Backup data to the repository Adding a backup snapshot is made with the following command:

```
borg create /path/to/repo::$(date +%Y%m%d-%H%M) <PATH_TO_DATA>
```

Listing backups stored in the repository Then, listing what backup has been made can be done with the following command:

```
borg list /path/to/repo
```

This command would return such result:

```
borg list /path/to/repo
20160902-1828                        Fri, 2016-09-02 18:28:33
20160902-1839                        Fri, 2016-09-02 18:39:01
20160902-2000                        Fri, 2016-09-02 20:00:01
20160902-2200                        Fri, 2016-09-02 22:00:01
20160903-0000                        Sat, 2016-09-03 00:00:01
20160903-0200                        Sat, 2016-09-03 02:00:01
20160903-0400                        Sat, 2016-09-03 04:00:01
```

Restoring data Then, restoring data can be made with the following command. The command must be executed in the folder where to restore files. Next example would restore **20160902-1828** backup listed in previous section.

```
borg extract /path/to/repo::20160902-1828
```

7.2. Troubleshooting

In order to troubleshoot TRE$_S$PASS platform, each container provides log through the following command, typed located in the root of the docker project, as mentioned in 6.3.2.2:

```
docker-compose logs <container_name>
```

This will display the logs of the command that the container runs. For instance, the CAS server will return log from Apache Tomcat:

```
cas_1       | ******************** Welcome to CAS ********************
cas_1       | CAS Version: 4.2.2
cas_1       | Build Date/Time: 2016-05-20T10:21:38.000Z
cas_1       | Java Home: /usr/lib/jvm/java-8-openjdk-amd64/jre
cas_1       | Java Vendor: Oracle Corporation
cas_1       | Java Version: 1.8.0_102
cas_1       | OS Architecture: amd64
cas_1       | OS Name: Linux
cas_1       | OS Version: 4.4.0-34-generic
cas_1       | **************************************************
cas_1       | >
cas_1       | 2016-09-02 14:02:32,752
cas_1       | INFO [org.jasig.cas.services.DefaultServices
```

7.3. Service restore procedure

In case of restore, we retrieve data from back-ups and follow the procedure described in section 6.3.2.

7.4. Manuals and documentation

7.4.1. Authentication process with CAS

All services, except the SVN, can be visited after only one authentication through CAS server. The session lasts as long as the browser is open.

When visiting for instance https://trespass-ticket.itrust.lu/, the user is forwarded to https://trespass-cas.itrust.lu where he/she authenticates.

If the authentication succeeds, the user is then forwarded to the https://trespass-ticket.itrust.lu/ website, which checks in the background (server side) with the CAS server who successfully authenticated. This enable the https://trespass-ticket.itrust.lu/ to get the user's identity and send him a newly created cookie.

Then, if the user now visits the https://trespass.itrust.lu website, he is forwarded to the CAS server for authentication, but as he already has a cookie corresponding to the CAS server, the latter forwards him directly back to the https://trespass.itrust.lu website, which also checks with CAS server the identity of the user in order to send him a session cookie.

(Apereo, 2016) presents the CAS protocol with a detailed web flow diagram.

7.4.2. Sign a jar file

In order to sign a jar, so it can be verified and executed by client, the certificate and key of signer must be used in form of a java keystore.

7.4.2.1. Requirements

In order to sign jar, please check following requirements:

- Have a java JDK in version 8 which include keytool and jarsigner commands;
- The certificate used to sign must have 'Code signing' enabled in the 'Extended Key Usage' attribute.

7.4.2.2. Convert pem file to keystore

Follow commands package .pem files to a keystore. Convert your pem files to pkcs12 format:

```
openssl pkcs12 -export \
               -in cert.pem \
               -inkey key.key \
               -out certkey.p12 \
               -name codesigner
```

Convert the pkcs12 file to a keystore:

```
keytool -importkeystore \
        -destkeystore keystore \
        -srckeystore certkey.p12 \
        -srcstoretype PKCS12 \
        -alias codesigner
```

7.4.2.3. Sign a jar

Use the keystore to sign a jar:

```
jarsigner -keystore keystore <JAR_FILE> codesigner
```

8. Conclusions

The final version of the prototype is the result of task T6.4 and reflects the integration, deployment and maintenance performed in the entire project. Tools developed by different work packages have been integrated in a platform and can be used to model and to analyse particular scenarios of the project case-studies. Tasks can be streamlined by the use of tool chains. Tools and toolchains can be run and accessed through an API.

The platform can be packed in a virtual machine allowing its use in industrial environments without network connections to the outside. So far, we consider the integration effort as successful as most of the available tools have been integrated. Validation was made by the case study team (WP7).

The integration team has been involved in tasks related to the the initial configuration and implementation of the platform and its user interface. The work on this task will continue until the end of the project on including the new tools and on increasing the level of integration between them.

The revised version of the tool handbook included throughout this deliverable acts as a manual for integration, deployment and maintenance of the tools platform. It describes the features that are available for users and administrators and how administrators can perform their tasks.

A. Project summary

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

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

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

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

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

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

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

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

The innovative visualisations (WP4) developed in TRE$_S$PASS focus particularly on visualising elements of the analysis, as this is key to the overall project goal of providing "decision support" to practitioners. However, visualisations contribute also to model development and data gathering. Practitioners can access the TREsPASS toolkit via the attack navigator map interface, which provides an intuitive means of selecting appropriate tools (WP6) for data gathering, modelling, analysis and visualisation. These can be used, individually or in combination, to strengthen operational and strategic decision-making.

A.1. Case studies

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

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

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

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

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

A.2. Overview of TRESPASS Integration

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

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

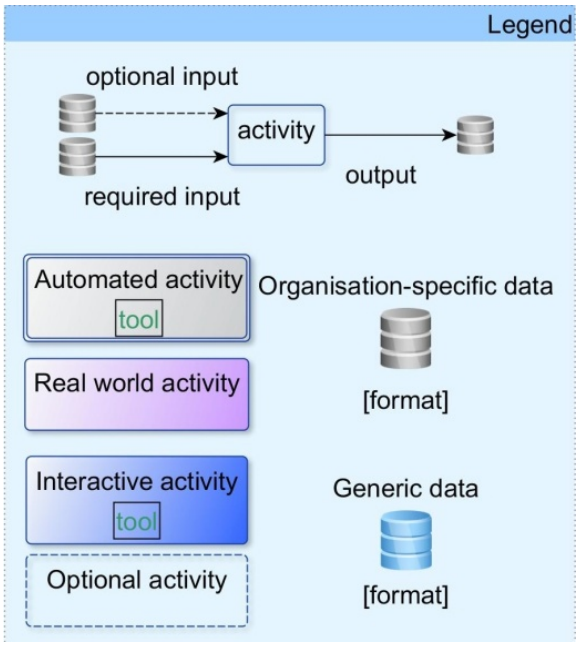

Figure A.1.: Legend for the Integration diagram.

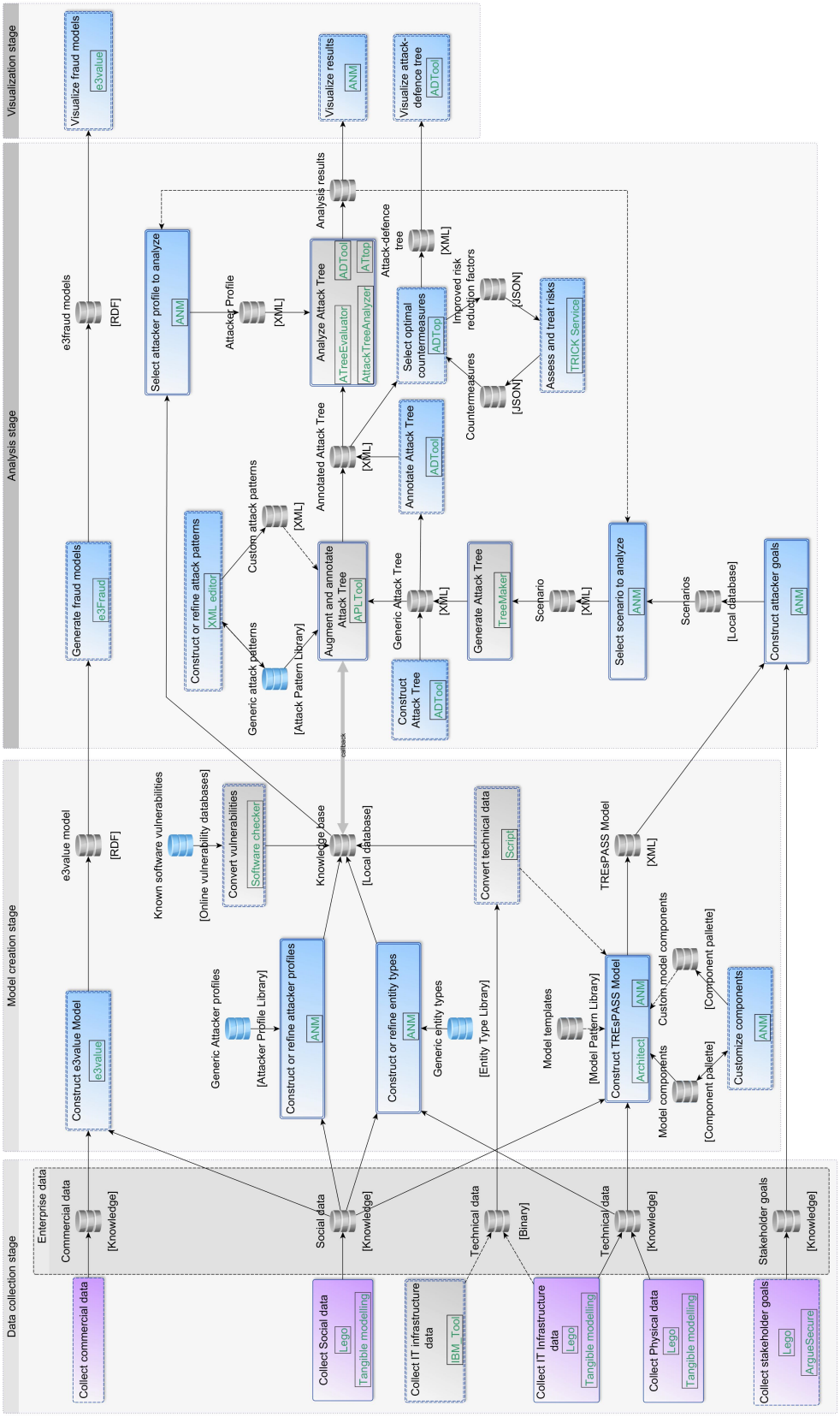

Figure A.2.: Integration diagram for the TRESPASS project.

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

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

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

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

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

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

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

Components customization (optional) takes place before or during the TRESPASS model creation to create specialized custom model components.

Attacker profile creation creates the attacker profile that the TRESPASS model analysis should consider, based on ready-made attacker profiles.

Defender/target profile creation creates similar profiles for the other actors in the model based on the social data gathered in the social data collection activity.

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

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

1. In the **attacker profile selection**, the user selects the attacker profile to use in the analysis.
2. The **attacker goals creation** provides the attack generation with the attacker goals. These can be derived by hand from the stakeholder goals or deduced automatically from the selected attacker profiles.
3. The **scenario selection** selects a scenario, consisting of a single pair of attacker and attacker goal, to run the TREsPASS analysis on.
4. To extend attack trees, **attack pattern creation and sharing** provides libraries with known attack steps. The attack tree generation can only reach a certain level of abstraction, which may not be sufficient for quantitative analyses.
5. **Attack generation** transforms the TREsPASS model to an attack tree.

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

6. **Attack tree annotation & augmentation** then extends the attack tree with attack patterns and decorates leaf nodes with parameter values from the data collection stage for quantitative analysis.
7. The **attack tree analyses** compute quantitative properties of attacks, e.g. utility for the attacker or success probability of the attack.

The analysis of the **e3value model** is complementary to the core TREsPASS analysis and has only one step: For the **fraud model generation**, the user needs select an attacker and an interval of expected occurrence rates of the commercial transactions specified by the e3value model. The e3fraud tool then identifies all possible violations of contracts, the loss for actors, and the delta in profit for the other actors.

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

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

B. Requirements for tool integration

Requirement R04

Requirement : Integration based on loose coupling and on a central integration component

Source WP: WP6

Target WP: WP6

Goals: Nature of data exchanges

Acceptance criteria: General integration approach, with flexibility to adapt to the tools being integrated

Status: Agreed and implemented

Dependencies: None

Comments: The different integration options with their advantages and drawbacks have been explained in (The TRESPASS Project, D6.1.2, 2015). Loose coupling is best suited for TRESPASS because both synchronous and asynchronous communications have to be supported (see Requirements R05.1 and R05.2). Tight coupling is mainly suited for synchronous exchanges, not for asynchronous exchanges. A central integration component reduces the number of interfaces to be realized, and makes it easier to secure the information exchanged between the tools.

Requirement R05.1

Requirement : Support for asynchronous data exchanges

Source WP: WP6

Target WP: WP6

Goals: Nature of tools from other WPs

Acceptance criteria: Tool integrated if requested

Status: Agreed

Dependencies: Partly conflicts R04

Comments: *Certain parts of the TRESPASS process require asynchronous communication between tools. E.g., data collection may take place prior to model creation, which means that the collected data needs to be stored first, or analysis may take place at a later stage than model creation.*

Requirement R05.2

Requirement : Support for synchronous data exchanges

Source WP: WP6

Target WP: WP6

Goals: At some point, there may be tools colecting and processing real time data. It will be implemented if needed

Acceptance criteria: Tool integrated if requested

Status: Agreed and implemented in ANM

Dependencies: None

Comments: *Certain parts of the TRESPASS process require synchronous communication between tools. E.g., in case of real-time visualisations of analysis results.*

Requirement R06

Requirement : "Apps style" user interface

Source WP: WP6

Target WP: WP6

Goals: Modern user-friendly style

Acceptance criteria: "Apps" should work stand alone as well as in a tool chain (whenever appllicable)

Status: Agreed and implemented

Dependencies: None

Comments: *Users of the TRESPASS toolset are not necessarily experts in the use of modelling and analysis tools. An intuitive, user-friendly user interface, consistent with other contemporary applications, makes the tools more accessible to these user groups.*

Requirement R10

Requirement : Interface between analysis tools and visualisation tools

Source WP: WP6

Target WP: WP4,6

Goals: Needed to generate visualisations

Acceptance criteria: The output of the analysis tools is available for visualisations

Status: Agreed and implemented

Dependencies: None

Requirement R12

Requirement : Develop a visual language

Source WP: WP4

Target WP: WP4,6

Goals: Needed for development of interface

Acceptance criteria: Documented language accepted by reviewers in Deliverable 4.1.1 (achieved) Accepted interpretation by practitioner panel and data analysis designers (WP) in Attack Navigator Map (iterative and on-going)

Status: Agreed and on-going

Dependencies: None

Requirement R13

Requirement : Define a visualisation process

Source WP: WP4

Target WP: WP4,5,6

Goals: Needed for development of interface

Acceptance criteria: Documented process accepted by reviewers in Deliverable 4.1.1 (achieved) Accepted interpretation by practitioner panel and data analysis designers (WP) in Attack Navigator Map (iterative and on-going)

Status: Agreed and on-going

Dependencies: None

Requirement R18

Requirement : Programmatic interface in other components in the TREsPASS tool

Source WP: WP1

Target WP: WP6

Goals: Needed to provide access to model

Acceptance criteria: The model is successfully integrated in a workflow ANM-model-treemaker-analyses-visualisations.

Status: Agreed and implemented

Dependencies: None

Requirement R40

Requirement : Support for attack tree visualisation

Source WP: WP3

Target WP: WP4,6

Goals: Improved presentation of tool output

Acceptance criteria: Visual inspection

Status: Agreed and implemented

Dependencies: None

Requirement R52

Requirement : Users can drag and drop standard elements onto the map.

Source WP: MT

Target WP: WP6

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 1 (Security Investment) and Use-case 2 (Audit), specifically U1.5, U2.4, U2.5, U4.1, U4.2, U5.6 and U5.7

Acceptance criteria: 80 percent of the users specified in the Use Cases are able to use the interface effectively after receiving the specified training. Standard elements must include business processes.

Status: Agreed and on-going

Dependencies: None

Requirement R53

Requirement : Users can add connections between elements on the map.

Source WP: MT

Target WP: WP6

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

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

Status: Agreed and on-going

Dependencies: Potentially conflicts R07, R38

Requirement R54

Requirement : Users can change parameters of map elements by clicking and entering new values

Source WP: MT

Target WP: WP6

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

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

Status: Agreed and on-going

Dependencies: Potentially conflicts R07, R38

Requirement R55

Requirement : Users can change parameters of map elements by clicking and requesting information from available data extraction tools.

Source WP: MT

Target WP: WP6

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

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

Status: Agreed and on-going

Dependencies: None

Requirement R56

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

Source WP: MT

Target WP: WP3,5,6

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

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

Status: Shelved

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

Requirement R57

Requirement : The user can select attacker profiles from the library.

Source WP: MT

Target WP: WP6

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 1 (Security Investment), specifically U1.9.

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

Status: Agreed and on-going

Dependencies: None

Requirement R58

Requirement : The user can build attacker profiles by editing their properties

Source WP: MT

Target WP: WP6

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 1 (Security Investment), specifically U1.6, U1.8 and U1.10

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

Status: Agreed and on-going

Dependencies: None

Requirement R59

Requirement : The user can assign asset values and properties to elements on the map.

Source WP: MT

Target WP: WP6

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

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

Status: Agreed and on-going

Dependencies: None

Requirement R60

Requirement : The user should be able to replicate an entity in the model by clicking on it, selecting the replicate option, and indicate the number of replications and associated parameters.

Source WP: MT

Target WP: WP6

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

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

Status: Agreed and on-going

Dependencies: None

Requirement R64

Requirement : The system should be able to handle push messages from external tools in order to generate push messages (updates) to users/models

Source WP: MT

Target WP: WP6

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 1 (Security Investment), specifically U1.1 and U1.13.

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

Status: Shelved

Dependencies: None

Requirement R68

Requirement : The TRESPASS tools enable the user to store a TRESPASS model, including navigator map, attacker profiles, and history of analyses

Source WP: MT

Target WP: WP6

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 1 (Security Investment), Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U1.18, U2.15, U4.3 and U5.17.

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

Status: Agreed and on-going

Dependencies: R7

Requirement R69

Requirement : The TRESPASS tools enable the user to load a previously stored TRESPASS model, including navigator map, attacker profiles, and history of analyses.

Source WP: MT

Target WP: WP6

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 4 (Product-service system), specifically U4.3

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

Status: Agreed and on-going

Dependencies: R7

Requirement R80

Requirement : The TRESPASS tools can start analyses automatically after updates of the corresponding models.

Source WP: MT

Target WP: WP6

Goals: Supports Use-case 1 (Security Investment), specifically U1.1

Acceptance criteria: Results of the chosen analyses are available after the corresponding events.

Status: Shelved

Dependencies: None

Requirement R81

Requirement : Visualisation of social and technical data, maps, scenarios, countermeasures.

Source WP: MT

Target WP: WP4,6

Goals: Derived from P7 (TRESPASS tools should be able to visualise maps, scenarios, and countermeasures). Supports Use-case 1 (Security Investment), Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U1.2, U2.6, U2.13, U4.6, U5.9 and U5.15.

Acceptance criteria: The visualisations are sufficiently expressive.

Status: Agreed and on-going

Dependencies: R03, R06

Requirement R82

Requirement : TRESPASS tools should be accessible through a web interface

Source WP: MT

Target WP: WP6

Goals:

Acceptance criteria: The web interface functions correctly on the most common platforms (Android, iOS).

Status: Agreed and implemented

Dependencies: R02, R06

Requirement R83

Requirement : The TRESPASS integrated component should use loosely coupled tools that are also available individually

Source WP: MT

Target WP: WP6

Goals:

Acceptance criteria: 80 percent of the users specified in the Use Cases are satisfied upon use of each individual tool separately.

Status: Agreed and implemented

Dependencies: R01

Requirement R86

Requirement : Users can select base models from a Model Template Library.

Source WP: MT

Target WP: WP4,5,6

Goals: Supports Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U4.2, U5.4

Acceptance criteria: 80 percent specified in the case studies are able to use the interface effectively after receiving the specified training.

Status: Agreed and on-going

Dependencies: Assumes the Model Template Library exists

Requirement R87

Requirement : Users can (re-) run analyses manually or at scheduled times/intervals

Source WP: MT

Target WP: WP6

Goals: Supports Use-case 2 (Audit) and Use-case 5 (Quick scan), specifically U2.14, U5.16

Acceptance criteria: 80 percent specified in the case studies are able to use the interface effectively after receiving the specified training.

Status: Shelved

Dependencies: None

C. List of tools

Tool name	Tool category	Tool description	Partner	Use case	Related WPs
ADTop	Analysis	This tool helps to describe a risk scenario using ADTrees and to automatically determine optimal sets of security controls. It uses the ROSI concept to evaluate the generated ADTrees, and finds the optimal ones.	ITR	Epstan Cloud	WP3.4.2 WP5.4.2 WP6.4.2 WP6.4.3 WP7.2.2 WP7.4.2
ATA	Analysis	Analyses if the considered threat model is feasible to rational profit-oriented adversaries. In the case of positive answer, provides the user with the list of top-10 most feasible attack vectors.	CYB	None	WP6.4
ATE	Analysis	The ATree Evaluator addresses multi-parameter optimisation of attack trees. The evaluation techniques characterise the leaves of a tree with more than one parameter, such as success probability and cost of an attack. Such multi-parameter optimisations are necessary in case of conflicting parameters, as there is no single best solution but rather a set of optimal solutions. We handle conflicting parameters by computing the set of efficient solutions, defined in terms of Pareto efficiency. The results computed by ATree Evaluator allow the defender for example to identify the maximum probability for a successfull attack, given an attacker budget.	DTU	Cloud	WP3
ATtop	Analysis	The ATtop provides two functionalities: (1) ATtop can input and output a variety of Attack Tree dialects; so it can be used to interchange Attack Trees between different tools by model transformation. (2) Depending on appropriate quantitative values in the basic attack steps, ATtop can analyse expected cost and time of attacks. It can compute expected cost under a time bound, expected time under a cost bound, or display the cumulative risk (probability x impact) over time. ATtop transforms Attack Trees to Priced Timed Automata, reusing UPPAAL as the underlying analysis engine.	UT	ATM	WP3
e3tool	Analysis	e3tool supports the construction of e3value value models as well as conducting profitability and fraud assessment of these models, based on the e3fraud methodology. An e3value model describes how economic value is created and exchanged within a network of actors. The e3fraud methodology consists of an ontological extension to e3value able to represent "sub-ideal" (i.e. fraudulent) value models, and a generation module which can identify, rank and visualise fraudulent variations of a given value model.	UT	Telco	WP7
Treemaker	Analysis	Treemaker identifies possible attacks in the model to reach an attack goal and generates attack trees for these. The approach to attack generation is based on policy invalidation on the socio-technical security model, and identifies ways of breaking policies in a system through recursive refinement of goals and identification of missing assets. The sequence of actions to identify missing assets and the actual actions performed is then translated into an attack tree.	DTU	All	WP1 WP3
TRICK Service	Analysis	TRICK Service assesses quantitative risks and fulfils risk treatment operations according to the user-configured referential standards (ISO/IEC 27001, 27002, etc.). It outputs the risk treatment plan which prioritises the implementation of security measures according to their Return On Security Investment (ROSI), risk specificities and feasibility.	ITR	Epstan Cloud	WP3.4.2 WP5.3.3 WP5.4.2 WP6.4.2 WP6.4.3 WP7.2.2 WP7.4.2
TREsPASS model	API	The TREsPASS model provides an abstraction layer for the data store. It is currently mainly used as interface between the information stored in the XML file and treemaker, as well as projects working directly on the model.	DTU	All	WP1
ArgueSecure-online	Data Collection	ArgueSecure is a set argumentation-based risk assessment tools aimed at documenting the rationale behind attacks identified and countermeasures selected as part of an informal, qualitative risk assessment: - A lightweight Excel-based version. - A Java version which is intended to be used during dedicated security requirements elicitation sessions. It is designed to be usable with a projector. - An online version which in addition allows stakeholders and experts to engage in a risk assessment in real-time without being in the same room and even without being available the same time.	UT	IPTV e-voting Cloud	WP1 WP5
InterActor	Data Collection	Users create private projects where initial problem-definitions are stated. Data from any number of separate workshops (including physical modelling) can be entered manually or imported as .csv or .json files. Using network graphs and a flow chart view the user can investigate and order the data, and construct a narrative from it for use with other tools.	RHUL	IPTV Others	WP4 WP2
SAVE	Data Collection	This tool is an IBM tool that automates the extraction of infrstructure details from virtualised environments. The tool supports a number of cloud technologies such as VMWare, OpenStack, KVM. The data is stored in an abstracted format that allows the formal modelling of isolation policies and properties.	IBM	Cloud	WP2
VMWare Extract	Data Collection	This tool extracts technical information from VMWare networks. The information includes infrastructure details and includes access control policies. The tool stores data in the TREsPASS defined format.	IBM	Cloud	WP2
ADTool	Design and visualisation of attack trees Analysis	The ADTool allows users to model and analyse attack-defense scenarios represented with attack-tree and attack-defense tree models. The ADTool implements many relevant attributes (attribute domains) for quantitative analysis, such as cost, satisfiability of a scenario, probability or an attack, and new attributes can be defined in the tool. The bottom-up computation algorithm is used for evaluating the selected attribute for the whole tree. The current stable release of the ADTool is able to import automatically generated attack trees from the APL, show top attack vectors in a tree (for some attributes), and it can be run in the scripting mode.	UL	ATM IPTV	WP3 WP4
APL	Misc	Decorates Treemaker output with attack patterns. Decorates attack tree leafs with quantitative annotations.	CYB	None	WP5
Software Checker	Misc	Determines if specific software versions contain any public vulnerabilities indexed by reliable databases such as CVE (http://nvd.nist.gov/).	ITR	None	WP6.2 WP6.3 WP6.4
Attack Navigator Map	Visualisation	The Attack Navigator Map (ANM) 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, with assets, items, actors, processes, policies and a set of attacker profiles. The analysis of the scenario is visualised to help make decisions.	LUST	All	WP1 WP2 WP3 WP4 WP5 WP6 WP7
Attack Tree Visualiser	Visualisation	The Attack Tree Visualiser is an online tool that visualises XML-files that can be loaded. The tool detects which flavour of Attack Tree-XML is used, shows AND and OR nodes and countermeasures. The result can be saved as PDF/SVG, or can be browed online by zooming in and out.	LUST	All	WP4
CEAV	Visualisation	The Cloud Environment Actor Visualiser (CEAV) visualises a cloud environment, including infrastructure like physical servers and virtual machines as well as cloud administrators, over time with a focus on the roles the administrators have on certain parts of the infrastructure. As cloud environments typically have a large number of components, the view focuses on the changing elements over a time interval and abstracts/summarises unchanging parts visually. The time interval of interest can be selected from a timeline that indicates changes in an overview of the available date range.	IBM	Cloud	WP4
TiCoVis	Visualisation	The Time-Containment Visualiser (TiCoVis) creates an 'alluvial' view of selected 'container-content' relations, e.g., between physical servers and virtual machines, over time. In an alluvial diagram, time is an integral part of the visualisation and the 'flow' of contained elements between containers will be visible over time. Zooming and panning functionality easily allows seeing the big picture over time as well as to show details for specific time intervals.	IBM	Cloud	WP4
Knowledge Base	Visualisation Analysis	The Knowledge Base serves as the TREsPASS Information System to gather all related input and configuration files that are required during the creation of a TREsPASS model for a given scenario and the run of analysis tools. It supports version control of all configuration and generated files, allowing a complete capture of the work on a specific scenario during all changes and re-runs. It is mainly used as the backend for the running of the ANM.	IBM	All	WP2 WP3 WP5

Table C.1.: List of tools developed in the context of the TRESPASS project (part 1).

Tool name	Reference to publication or TREsPASS document	Value
ADTop	Master's thesis - Cédric Muller "Approach to automated selection of preventive security controls in attack-defence trees". Springer paper (GraMSec'16) - "Bridging Two Worlds: Reconciling Practical Risk Assessment Methodologies with Theory of Attack Trees". Fetler B., Hapes C. - "Information Security Maturity as an Integral Part of ISMS based Risk Management Tools". D.7.2.2. Use case for cloud computing (with attack tree coming from the deliverable D.7.2.1). D.7.4.2. Epstan use case for Privacy.	Helps to provide complementary proposals for risk analyses and security implementation of countermeasures in an organisation's information system, with results based on ROSI optimisation and attack-defence trees.
ATA	-	Helps to assess the feasibility of attacking for the rational profit-oriented adversaries.
ATE	Aslanyan, Z., & Nielson, F.: Pareto efficient solutions of attack-defence trees. In Principles of security and trust - 4th international conference, POST 2015, D3.3.2.	Analyse conflicting values, relate incomparable parameters.
ATtop	R.Kumar, E. Ruitjers, M. Stoelinga: "Quantitative Attack Tree Analysis via Priced Timed Automata" in FORMATS 2015. D3.3.2; D3.2.1.	Bridge the Attack Tree formats / limitations of various tools. Compute probability and expected costs of attacks as a function over time.
e3tool	D. Ionita, S. Koenen, and R. Wieringa, "Modelling telecom fraud with e3value," 2014. URL: http://doc.utwente.nl/92422/. Dan Ionita, Roel J. Wieringa, Lars Wolos, Jaap Gordijn and Wolter Pieters. Using value models for business risk analysis in e-service networks. In Jolita Ralyté, Sergio España and Óscar Pastor editors, The Practice of Enterprise Modeling, Pages 239–253, Springer International Publishing, 2015. Dan Ionita, Roel J. Wieringa and Jaap Gordijn. Automated Identification and Prioritization of Business Risks in e-service Networks. In Theodor Borangiu, Monica Dragoicea and Henriqueta N'ovoa editors, Exploring Services Science: 7th International Conference, IESS 2016, Bucharest, Romania, May 25-27, 2016, Proceedings, Pages 547–560, Springer International Publishing, Cham, 2016. Dan Ionita, Jaap Gordijn, Ahmed Seid Yesuf and Roel J. Wieringa, Value-driven risk analysis of coordination models in The Practice of Enterprise Modeling, Springer International Publishing, 2016. All D7.3.x deliverables.	Speeds up fraud analysis of new e-services by allowing automated identification and analysis of business risks based on know heuristics as well as analysis of manually constructed fraud scenarios.
Treemaker	Kammüller, F., & Probst, C. (2013). Invalidating policies using structural information. In 2nd international ieee workshop on research on insider threats, writ'13. Marieta Georgieva Ivanova, Christian W. Probst, René Rydhof Hansen, Florian Kammüller: Transforming Graphical System Models to Graphical Attack Models. GraMSec@CSF 2015 Marieta Georgieva Ivanova, Christian W. Probst, René Rydhof Hansen, Florian Kammüller: Attack Tree Generation by Policy Invalidation. WISTP 2015: 249-259 Florian Kammüller, Christian W. Probst: Invalidating Policies using Structural Information. JoWUA 5(2): 59-79 (2014) D3.4.1.	Provides the input for the analyses.
TRICK Service	D6.4.2, D.6.4.3., D.7.4.2., D.7.4.4. used for risk analysis.	Interface to perform a risk analysis and compute an action plan for an organisation, based on catalogues/standards for standardization and security controls.
TREsPASS model	Marieta Georgieva Ivanova, Christian W. Probst, René Rydhof Hansen, Florian Kammüller: Transforming Graphical System Models to Graphical Attack Models. GraMSec@CSF 2015 D1.3.1; D1.2.2.	Provides interface to information in XML file.
ArgueSecure-online	D. Ionita, R. Kegel, R. J. Wieringa, and A. Baltuta, "ArgueSecure: Out-of-the-Box Security Risk Assessment," in Evolving Security and Privacy Requirements Engineering (ESPRE) Workshop, co-located with the 24th IEEE International Requirements Engineering Conference, Beijing, China, 2016. D. Ionita, J. W. Bullee and R. J. Wieringa, "Argumentation-based security requirements elicitation: The next round," Evolving Security and Privacy Requirements Engineering (ESPRE), 2014 IEEE 1st Workshop on, Karlskrona, 2014, pp. 7-12. doi: 10.1109/ESPRE.2014.6890521, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6890521&isnumber=6890516 Henry Prakken, Dan Ionita, and Roel Wieringa. 2013. Risk Assessment as an Argumentation Game. In Proceedings of the 14th International Workshop on Computational Logic in Multi-Agent Systems - Volume 8143 (CLIMA XIV), João Leite, Tran Son, Paolo Torroni, Leon Torre, and Stefan Woltran (Eds.), Vol. 8143. Springer-Verlag New York, Inc., New York, NY, USA, 357-373. DOI: http://dx.doi.org/10.1007/978-3-642-40624-9_22	Quick, intuitive, qualitative RA that can be used when no model of the system or quantiative estimations of impact or likelihood are available or reasonably obtainable. Can also be used as a precursor to a full-fledged RA in order to gain an idea of stakeholders' perpective on risks and improtant assets.
InterActor	D4.3.3 plus the following paper on the LEGO method: Logical Lego?: Co-constructed perspectives on service design. NordDesign 2014, presented at Swinburne Institute of Technology, Melbourne. Authors: Claude P.R. Heath, Lizzie Coles-Kemp, Peter A. Hall. And papers on the practice of visualsiation, including LEGO: 'Visualisation in Cyber-security: Towards a Critical Practice.' EVAA 2016 (Electronic Visualisation and the Arts Australasia). Presented in Canberra. Authors: Peter A. Hall, Lizzie Coles-Kemp, Claude P.R. Heath. 'Critical visualization: a case for rethinking how we visualize risk and security.' Journal of Journal of Cybersecurity 2015, p.116. Inaugural Issue. Authors: Peter A. Hall, Claude P.R. Heath, Lizzie Coles-Kemp. 'Critical Visualisation: a case for rethinking how we visualise risk and security.' New Security Paradigms Workshop (NSPW) 2015. Presented at Twente University, Netherlands. Authors: Claude P.R. Heath, Lizzie Coles-Kemp, Peter A. Hall. 'The Navigation Metaphor in Security Economics.' IEEE Security & Special Issue: Economics of Cyber-security. Presented at Twente University, Netherlands. Authors: Wolter Pieters, Margaret Ford, Claude P.R. Heath, Christian W. Probst, Ruud Verbij.	The app, in conjuction with the physical modelling methodology using LEGO, can be used to transcribe key elements of the physical models into digital form. Diverse forms of participatory brainstorming and co-creation activities can thus be extended into working with shareable data and visualising it in a number of ways.
SAVE	-	Tool for automating the extraction of complex cloud infrastructures.
VMWare Extract	-	Tool for automating the extraction of complex cloud infrastructures and access control policies.
ADTool	B. Kordy, P. Kordy, S. Mauw, P. Schweitzer: "ADTool: Security Analysis with Attack-Defense Trees" in QEST 2013 O. Gadyatskaya, R. Jhawar, P. Kordy, K. Lounis, S. Mauw, R. Trujillo-Rasua "Attack Trees for Practical Security Assessment: Ranking of Attack Scenarios with ADTool 2.0" in QEST 2016. D3.1.1; D3.4.2	Open source cross-platform tool to quickly draw attack trees and quantitative evaluate attack scenarios.
APL	-	Enables quantitative attack tree analysis
Software Checker	-	Software solution for supervising software installed on a set of Windows clients to ensure that all tools and applications installed on a computer are up-to-date in order to avoid vulnerabilities.
Attack Navigator Map	D6.3.1.	Connects many tools developed in the project, works in a visual manner, bridges between model and analysis, through visualisation of results. Runs in a browser without plug-ins.
Attack Tree Visualiser	D4.2.2.	Immediate visualisation of XML files.
CEAV	D4.3.2.	Visualisation of large cloud structure (actor - role - cloud infrastructure) over time.
TiCoVis	D4.3.2.	Visualisation of 'container - content'-related large data sets over time.
Knowledge Base	D2.4.1.	Backend for ANM handling tool runs

Table C.2.: List of tools developed in the context of the TRESPASS project (part 2).

Tool name	Input/output	Tool location	Input files	External link (URL)
ADTop	Input: ATree (XML file), extract of risk analysis (JSON file). Output: Optimal ADTree (XML file). Input/Output: Association Matrix (XLS file).	https://trespass-svn.itrust.lu/ADTop TREsPASS portal: https://trespass.itrust.lu	/input/ADTop/Association matrix.xls /input/ADTop/ATree.xml /input/ADTop/extractTS.json	-
ATA	Input: the threat model in the form of a monotone Boolean function. Variables of the function are annotated with cost-success ratios (mean cost for attacking). Output: feasibility of attacking for rational profit-oriented adversaries. A list of top-10 feasible attack vectors, if attacking is feasible.	https://trespass-svn.itrust.lu /attack_tree_analyzer	/input/ATA/tree.xml /input/ATA/FF.properties /input/ATA/profileV.xml	-
ATE	Input: ATree (XML file). Output: analysis results.	https://trespass-svn.itrust.lu/ATE/	-	-
ATtop	Input: ATree (XML file). Output: ATtree (XML file), UPPAAL model (XML file), or analysis results.	https://trespass-svn.itrust.lu/ATtop/	/input/ATtop/ATTop%20instructions.docx /input/ATtop/EnterBuilding.xml /input/ATtop/StuxNet.at	-
e3tool	N/A	https://github.com/danionita/e3tools	-	https://github.com/danionita/e3tools/releases
Treemaker	Input: model file (XML file). Output: adtree file	tool https://trespass-svn.itrust.lu/treemaker/ source https://www.trespass-project.eu/repositories /TREsPASS/trunk/PartnerSpaces/DTU/tools/treemaker/	https://www.trespass-project.eu/repositories /TREsPASS/trunk/PartnerSpaces/DTU/tools/treemaker/models /input/Treemaker/input.zip	-
TRICK Service	Output: extract of risk analysis (JSON file), requires a connection to the TRICK Service API when using ADTop.	TREsPASS portal: https://trespass.itrust.lu	-	-
TREsPASS model	Input: model file (XML file).	source https://www.trespass-project.eu/repositories /TREsPASS/trunk/PartnerSpaces/DTU/tools/treemaker/	-	-
ArgueSecure-online	N/A	https://danionita.github.io/ArgueSecure/	-	ArgueSecure.ewi.utwente.nl
InterActor	Input: physical and other forms of modelling. Output: XML, exported .csv files, PDFs of graphs and other visualisations of actor networks.	https://trespass-svn.itrust.lu/INTERActor/	-	-
SAVE	Input: None. Output: Infrastructure details in graph format.	-	-	-
VMWare Extract	Input: None. Output: Infrastructure and policy details.	-	-	-
ADTool	Input: (optional) attack tree (XML file). Output: (optional) evaluated/modified attack tree (XML file); figure with the attack tree (png, pdf).	https://github.com/tahti/ADTool2	-	-
APL	Input: treemaker output Output: decorated and annotated attack tree	https://trespass-svn.itrust.lu/AttackPatternLibrary	-	-
Software Checker	Input: Input file for testing (tab separated), NVD	TREsPASS portal: https://trespass.itrust.lu	/input/SoftwareChecker/Software.txt	http://nvd.nist.gov
Attack Navigator Map	Input (optional): model file (XML). Output: scenario file + model file.	https://trespass.itrust.lu/attack-navigator-map	-	-
Attack Tree Visualiser	XML/PDF.	TBD /TREsPASS platform.	Url TBD.	-
CEAV	Part of running instance.	To be integrated into ANM/KB code.	Part of running instance on itrust site (url https://trespass-tkb.itrust.lu/tkb/CEAV).	-
TiCoVis	Part of running instance.	To be integrated into ANM/KB code.	Part of running instance on itrust site (url https://trespass-tkb.itrust.lu/tkb/TiCoVis).	-
Knowledge Base	No specific inputs beyond the different required for the various tools called (i.e., ANM, treemaker, APL, ATA, ATE).	https://trespass-project.eu/repositories /TREsPASS/trunk/PartnerSpaces/IBM/TREsPASS_KB/ https://trespass-svn.itrust.lu/TREsPASS_KB/	Part of running instance at https://trespass-tkb.itrust.lu/anm/	-

Table C.3.: List of tools developed in the context of the TRE$_S$PASS project (part 3).

Tool name	Where in the map?	Implementation	Next steps	Intellectual property rights	Maturity (TRL)
ADTop	TRICK Service -> ADTop -> ADTool ADTool -> ADTop -> ADTool	Desktop application Java	Sort optimal ADTree based on different parameters (cost, ROSI, success probability), apply heuristics to accellerate the computation, facilitate automation of association matrix thanks to libraries (optional).	Copyright itrust consulting, all rights reserved. The tool will be published open-source, with the clause-3 BSD license. TREsPASS portal will also be published open-source with Apache 2 license.	5
ATA	APL -> ATA -> Visualisation	Command line application C++	-	Copyright Cybernetica, all rights reserved.	9
ATE	Analysis tools.	Java	-	-	5
ATtop	Analysis tools.	Java	-	-	5
e3tool	Separate fork (top of diagram).	Desktop application Java	Testing outside telco case study in order to obtain new heuristics.	Open-source GPLv3.	5
Treemaker	Between model and analysis.	Scala	-	-	5
TRICK Service	TRICK Service -> ADTop	Web application Java/html/javascript	TRICK Service will be upgraded with dynamic parameters for risk monitoring and may contain other features linked to attack-defence trees or coming from requests of customers.	Copyright itrust consulting, all rights reserved.	9
TREsPASS model	Model.	Scala	-	-	5
ArgueSecure-online	Data collection (stakeholder goals).	Desktop application (Java) Online browser-based Excel-implementation	None.	Open-source GPLv3.	5
InterActor	Input to ANM, augments the ANM with a view of an array of social data relevant to the scenarios.	-	Final rounds of user testing and feedback will be incorporated into the last stages of tool development (Sept 2016).	Open source.	5
SAVE	-	-	-	-	-
VMWare Extract	-	-	-	-	-
ADTool	Analysis and visualisation tools.	Desktop application Java	Bug fixing (within TREsPASS).	Open source.	6
APL	Treemaker -> APL -> [Analysis tools]	Python	-	-	5
Software Checker	Output: Knowledge base (local database) Software Checker -> Knowledge base	Command line application Java/html/javascript	-	Copyright itrust consulting, all rights reserved.	5
Attack Navigator Map	Spans model creation, analysis, and visualisation.	javascript	Bug fixing (within TREsPASS).	Open source (soon).	4
Attack Tree Visualiser	Visualisation.	html/javascript	-	-	-
CEAV	Visualisation.	html/javascript	-	-	-
TiCoVis	Visualisation.	html/javascript	-	-	-
Knowledge Base	Spans model creation, analysis, and visualisation.	html/javascript/python	-	-	-

Table C.4.: List of tools developed in the context of the TRE$_S$PASS project (part 4).

Technology readiness levels (TRL). Where a topic description refers to a TRL, the following definitions apply, unless otherwise specified:
TRL 1 – basic principles observed.
TRL 2 – technology concept formulated.
TRL 3 – experimental proof of concept.
TRL 4 – technology validated in lab.
TRL 5 – technology validated in relevant environment (industrially relevant environment in the case of key enabling technologies).
TRL 6 – technology demonstrated in relevant environment (industrially relevant environment in the case of key enabling technologies).
TRL 7 – system prototype demonstration in operational environment.
TRL 8 – system complete and qualified.
TRL 9 – actual system proven in operational environment (competitive manufacturing in the case of key enabling technologies; or in space).

Table C.5.: Maturity (TRL).

Tool name	Limitations (e.g. only works for x scenario or dataset)	Exploitation	Dissemination (e.g. published, used by company X, or referenced by research X	User requirements status	Additional information for applications to be executed on the server side (and not client), e.g. Treemaker.	Comments
ADTop	Currently limited to 17 countermeasures due to memory management and initial algorithm design, but improvements are planned.	Planned to be used in EpStan project.	Used by research partners and security practioners. References to the tool described in column H.	Completed	-	User needs to have Java installed on his/her computer.
ATA	-	-	-	Completed	-	Requires Java SE 8. May require adversarial profile specification, as well as internal heuristic settings (depending on the chosen mode of operation)
ATE	-	-	-	-	Based on treemaker/APL generated output.	-
ATtop	-	-	-	-	Needs UPPAAL "verifyta" tool in the PATH or in the working directory (UPPAAL provided in the ATtop directory).	-
e3tool	https://github.com/danionita/e3tools/issues	-	Used by D-Telekom and published.	Partly completed	Cannot be executed server-side.	Requires Java
Treemaker	-	-	-	-	-	-
TRICK Service	None known.	Commercial tool for security practitioners. Licence sold to customers.	Used by customers in Luxembourg.	Completed	-	-
TREsPASS model	-	-	-	-	-	-
ArgueSecure-online	None known.	-	Published.	Completed	Available on the respective GitHub pages for the online version.	-
InterActor	None known.	Tool to be promoted and disseminated via Governmental and business supporters, for use by security practitioners and consultants.	Multiple workshops at RHUL, social media, and RSD5 (Relating Systems Thinking and Design Symposium), Toronto, October 2016. The LEGO method was also given in a Masterclass at CyberUK in 2016, in Liverpool, UK, sponsored by CESG and GCHQ.	Completed	-	Enables fine grain analysis of transcripts of participant speech, and other photographic and ordinal data. The tool also allows for high-level narrative segments to be gathered into a Sankey-style flow view. These two approaches can also be combined.
SAVE	-	-	-	-	-	-
VMWare Extract	-	-	-	-	-	-
ADTool	-	-	Published; referenced by many research papers.	Completed	-	Requires Java
APL	-	-	-	Completed	-	Requires python 3.0
Software Checker	None known.	Licence sold to customers.	Used by customers in Luxembourg.	Completed	-	-
Attack Navigator Map	Gets harder to use the bigger the scenario is.	By TREsPASS partners.	D6.3.1. Pieters W., Barendse J., Ford M., Heath C.PR, Probst C.W, Verbij R.. 2016. The Navigation Metaphor in Security Economics. IEEE Security & Privacy. 14:14–21. DOI: 10.1109/MSP.2016.47	Completed	All analysis tools run server side as well as the tool chains.	-
Attack Tree Visualiser	-	-	-	-	-	-
CEAV	-	-	-	-	-	-
TiCoVis	-	-	-	-	-	-
Knowledge Base	-	-	-	-	-	-

Table C.6.: List of tools developed in the context of the TRE$_S$PASS project (part 5).

Tool name	Tool evaluation method (what has been done, by whom, how many subjects, etc.)	Tool evaluation results (data)	Tool evaluation discussion (what the data means)
ADTop	Demonstration in Lisbon for GraMSec'16 to panel of security analysts, researchers and teachers from the University of Luxembourg, TREsPASS partners. Bug fixing ongoing, development phase.	Optimal attack-defence trees (XML files) based on ROSI optimisation. Results obtained for cloud computing and Epstan use cases. Risk reduction factors (JSON file) as complementary data (non-linear model).	Allows to compare RRFs with the ones of TRICK Service (linear model). Provide more details on attack scenarios and risk reduction to users. Allows to provide final implementation overview of an information system and various proposals for countermeasure implementation to users.
ATA	-	-	-
ATE	-	-	-
ATtop	-	-	-
e3tool	Iteratively evaluated by researchers at GUF and practitioners at D-Telekom.	To be published.	To be published.
Treemaker	-	-	-
TRICK Service	Tested for several clients, and continuous improvement thanks to feedback from users (customers, security practitioners).	Risk analyses conducted for our customers.	Real word data reflecting current information systems of customers.
TREsPASS model	-	-	-
ArgueSecure-online	Spreadsheet version: focus group with information security researchers at the UT and focus group with security consultants/researchers at Cybernetica. Offline version: a pilot study within PISA project, a focus group in a major Dutch bank and on ATM case study. Online version: observational live study during REFSQ 2016 conference and a focus group with information security researchers at the UT.	See "ArgueSecure: Out-of-the-Box Security Risk Assessment" paper.	See "ArgueSecure: Out-of-the-Box Security Risk Assessment" paper.
InterActor	The tool has emerged from multiple user engagements using the TREsPASS LEGO method with security practitioners and members of the public since 2014, examining a wide variety of scenarios. The total number of participants involved in this is 174, many of whom have given detailed feedback and suggestions for how the method might be extended.	Data from evaluations will be included in final Deliverables and placed in Repository on completion. This will be in the form of completed questionnaires, photographic documentation and audio and video recordings. This data will also be published in due course.	Feedback from the use of the tool by practitioners should confirm its usefulness in the workplace, and all feedback points (both postive and negative) will be addressed through the continued development of the tool and through subsequent written feedback to our participants.
SAVE	The tool has been evaluated by a selected group of IBM customers using a demonstration followed by a survey.	See D7.2.2.	-
VMWare Extract	The tool has been evaluated by internal cloud experts using interviews	-	-
ADTool	Tool has been used in the ATM and IPTV case studies (groups of security analysis used the ADTool to design attack trees).	-	-
APL	-	-	-
Software Checker	Tested internally by itrust consulting, then deployed on several worstations at the customer's premises.	The software is installed on computers.	It gives the list of potential security vulnerabilities due to obsolete versions of tools/applications installed on these computers.
Attack Navigator Map	Tested by TREsPASS members, in several feedback panels at various stages of the project. As part of user journeys by RHUL.	Descriptions of results can be found in D6.3.1. and D4.2.2.	Evaluation points have been used to improve the tool and add functonality to make it fit to the expected users.
Attack Tree Visualiser	The visualiation output of the tool is based on various feedback panels with security practioners.	-	-
CEAV	With IBM cloud admins.	-	-
TiCoVis	With IBM cloud admins.	-	-
Knowledge Base	Together with ANM.	-	-

Table C.7.: List of tools developed in the context of the TRE$_S$PASS project (part 6).

References

Apereo. (2016). *Cas protocol.* Retrieved 2016-04-16, from https://apereo.github.io/cas/4.2.x/protocol/CAS-Protocol.html

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

The TRESPASS Project, D2.2.1. (2013). *Technical data extraction prototype.* (Deliverable D2.2.1)

The TRESPASS Project, D2.4.1. (2016). *TRESPASS information system.* (Deliverable D2.4.1)

The TRESPASS Project, D3.3.1. (2013). *First report on stochastic analysis methods.* (Deliverable D3.3.1)

The TRESPASS Project, D3.4.2. (2016). *Methods for attack generation, preventive measures, and ranking.* (Deliverable D3.4.2)

The TRESPASS Project, D4.1.1. (2013). *Initial requirements for visualisation processes and tools.* (Deliverable D4.1.1)

The TRESPASS Project, D5.3.1. (2013). *Abstraction levels for model sharing.* (Deliverable D5.3.1)

The TRESPASS Project, D5.3.3. (2016). *Best practices for model maintenance.* (Deliverable D5.3.3)

The TRESPASS Project, D5.4.2. (2016). *The integrated TRESPASS process.* (Deliverable D5.4.2)

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, D7.2.2. (2016). *Final report case study a.* (Deliverable D7.2.2)

The TRESPASS Project, D7.4.2. (2016). *Final report case study c.* (Deliverable D7.4.2)