predict
prioritise
prevent

TRESPASS

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

Deliverable D6.4.4

The integrated TRESPASS tools

Project:	TRESPASS
Project Number:	ICT-318003
Deliverable:	D6.4.4
Title:	The integrated TRESPASS tools
Version:	1.0
Confidentiality:	Public
Editor:	Cédric Muller
Cont. Authors:	B. Jager, C. Muller, C. Harpes
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	Benoît Jager, Cédric Muller, Carlo Harpes	ALL

Quality assurance		
Role	Name	Date
Editor	Cédric Muller (ITR)	2016-10-31
Reviewer	Mariëlle Stoelinga (UT)	2016-10-18
Reviewer	Christian W. Probst (DTU)	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	2016-10-10
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

Management summary viii

1. Introduction 1
1.1. Goals of the deliverable . . . 1
1.2. Foreground and background . . . 1
1.3. Structure of the document . . . 1

2. TRESPASS integrated tools 2
2.1. Execution of the tools . . . 2
2.2. Methodology of the tool validation . . . 3
2.3. Overview of the test results . . . 3

3. List of applications 5
3.1. Treemaker . . . 5
3.1.1. Description . . . 5
3.1.2. Input . . . 5
3.1.3. Output . . . 6
3.2. Software Checker . . . 6
3.2.1. Description . . . 6
3.2.2. Input . . . 6
3.2.3. Output . . . 7
3.3. Attack Tree Analyzer (ATA) . . . 7
3.3.1. Description . . . 7
3.3.2. Input . . . 7
3.3.3. Output . . . 7
3.4. Attack Pattern Library (APL) . . . 8
3.4.1. Description . . . 8
3.4.2. Input . . . 8
3.4.3. Output . . . 8
3.5. RawDataProcessing . . . 8
3.5.1. Description . . . 8
3.5.2. Input . . . 8
3.5.3. Output . . . 8
3.6. TRESPASS Model . . . 9
3.6.1. Description . . . 9
3.6.2. Input . . . 9
3.6.3. Output . . . 9

3.7. Attack Tree Optimisation tool (ATtop) . . . 9
3.7.1. Description . . . 9
3.7.2. Input . . . 10
3.7.3. Output . . . 10
3.7.4. Usage . . . 10

4. List of tools **12**
4.1. Attack–Defense Tree Tool (ADTool) . . . 12
4.1.1. Description . . . 12
4.1.2. Connection . . . 13
4.1.3. Usage . . . 15
4.2. Attack Navigator Map (ANM) . . . 15
4.2.1. Description . . . 15
4.2.2. Connection . . . 16
4.2.3. Usage . . . 17
4.3. ArgueSecure online . . . 17
4.3.1. Description . . . 17
4.3.2. Usage . . . 18
4.4. ArgueSecure offline . . . 22
4.4.1. Description . . . 22
4.4.2. Usage . . . 24
4.5. ArgueSecure spreadsheets . . . 25
4.5.1. Description . . . 25
4.5.2. Usage . . . 26
4.6. TRICK Service . . . 27
4.6.1. Description . . . 27
4.6.2. Connection . . . 27
4.6.3. Usage . . . 28
4.7. InterActor . . . 28
4.7.1. Description . . . 28
4.7.2. Connection . . . 29
4.7.3. Usage . . . 31
4.8. Architect . . . 33
4.8.1. Description . . . 33
4.8.2. Connection . . . 34
4.8.3. Usage . . . 35
4.9. Attack Tree Evaluator (ATE) . . . 36
4.9.1. Description . . . 36
4.9.2. Input . . . 36
4.9.3. Output . . . 36
4.9.4. Connection . . . 36
4.9.5. Usage . . . 37
4.10. e3tool . . . 38
4.10.1. Description . . . 38
4.10.2. Input . . . 39
4.10.3. Output . . . 39

4.10.4. Connection 39
4.10.5. Usage 39
4.11. Attack-Defence Tree optimiser (ADTop) 43
4.11.1. Description 43
4.11.2. Connection 43
4.11.3. Usage 44

5. Validation of toolchains **46**
5.1. What worked 46
5.2. What didn't 46

6. Conclusions **47**

A. Integration diagram **48**

References **50**

List of Figures

2.1. File selector screen of ATtop as example for starting a tool. 2
2.2. Example of an executed task. 3

4.1. Example of execution warning message. 14
4.2. ADTool. 15
4.3. ANM home page. 17
4.4. Available assessments. 19
4.5. Create new assessment. 20
4.6. Example of exported assessment. 21
4.7. ArgueSecure offline. 23
4.8. ArgueSecure spreadsheets. 26
4.9. TRICK Service home page. 28
4.10. Register to InterActor. 29
4.11. Login to InterActor. 29
4.12. Welcome page of InterActor. 30
4.13. List of projects. 31
4.14. Edit a project. 32
4.15. Demo workshop. 33
4.16. Connection to Architect. 34
4.17. Open remote access to server. 34
4.18. Opening Architect. 35
4.19. Welcome page of Architect. 35
4.20. Attack Tree Evaluator. 37
4.21. e3tool editor. 40
4.22. Example of an e3tool value model. 41
4.23. e3tool: choose main actor. 41
4.24. e3tool: choose a parameter. 42
4.25. e3tool: enter occurrences. 42
4.26. e3tool: profitability chart. 42
4.27. ADTop. 44
4.28. ADTop: Optimal attack-defence tree provided. 45

A.1. Legend for the Integration diagram. 48
A.2. Integration diagram for the TRE$_S$PASS project. 49

List of Tables

2.1. List of applications. 3
2.2. List of tools. 4

Management summary

Key takeaways:

- A set of seven applications hosted and executed on the TRE$_S$PASS platform and eleven tools (dowloadable or linking to other web services) have been integrated in the TRE$_S$PASS platform;
- This report documents:
  - how the tools have been tested;
  - how the applications produce on a given input the expected output requested by the tool developer.
- The toolchain process has been checked successfully with the applications;
- The tools have been tested individually.

1. Introduction

1.1. Goals of the deliverable

This document is the public deliverable D6.4.4 *The integrated TRESPASS tools*.

According to the project Description of Work, this document reports on the final version of the integrated TRESPASS tools and is a demonstration of the tools as described in the tool handbook. This deliverable, together with D5.4.2 (The TRESPASS Project, D5.4.2, 2016) describe the TRESPASS process and tools, the main contribution of TRESPASS. It documents tests executed on the TRESPASS platform performed during roughly one week to make sure that all tools and applications work properly.

The revised version of the tool handbook is included throughout the D6.4.3 deliverable (The TRESPASS Project, D6.4.3, 2016). D6.4.3 also describes the TRESPASS user interfaces and provides a user and management guide for the TRESPASS platform, including its integration, deployment, and maintenance.

1.2. Foreground and background

The various percentages about foreground, background and sideground of the tools and applications developed during the TRESPASS project time frame are available in the deliverable D8.3.2 (The TRESPASS Project, D8.3.2, 2016).

1.3. Structure of the document

The document is structured as follows: chapter 2 provides the reader about the procedure to execute a tool or application and the methodology that has been followed, chapter 3 and 4 show the demonstration of the integrated TRESPASS tools and applications as described in D6.4.3, chapter 5 shows an example for the validation of a tool chain, and chapter 6 concludes this document.

2. TRESPASS integrated tools

2.1. Execution of the tools

Once connected to the TRESPASS platform, and once opened the "Tools" menu (see D6.4.3 for additional information) users can execute any tool in the following way:

1. Click on the run icon of the tool (located in the top-right corner of the tool box);
2. On the new loaded page, use the file selector to upload the input file(s) required for the selected tool, as shown in Figure 2.1. The browse button has the name of the required file, e.g. "input" if the required file is called input.zip.
3. Click on upload;

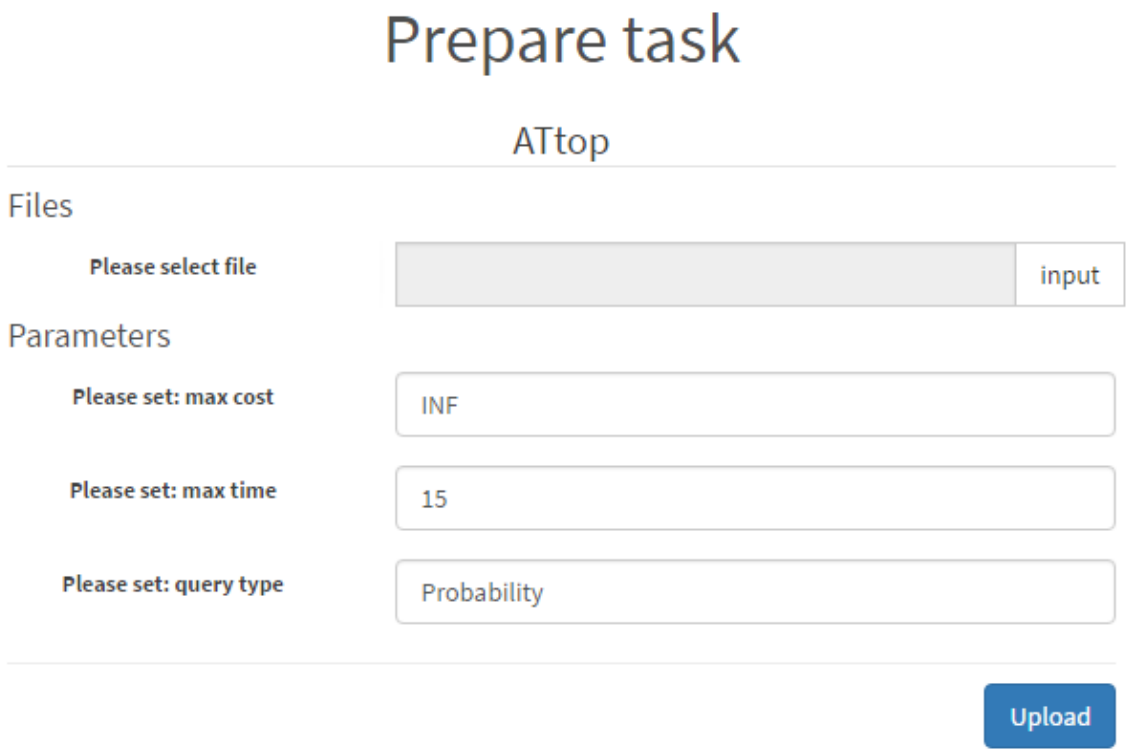

Figure 2.1.: File selector screen of ATtop as example for starting a tool.

4. It opens a new window which contains the executed tasks (the last run box appears, for instance as shown in Figure 2.2).

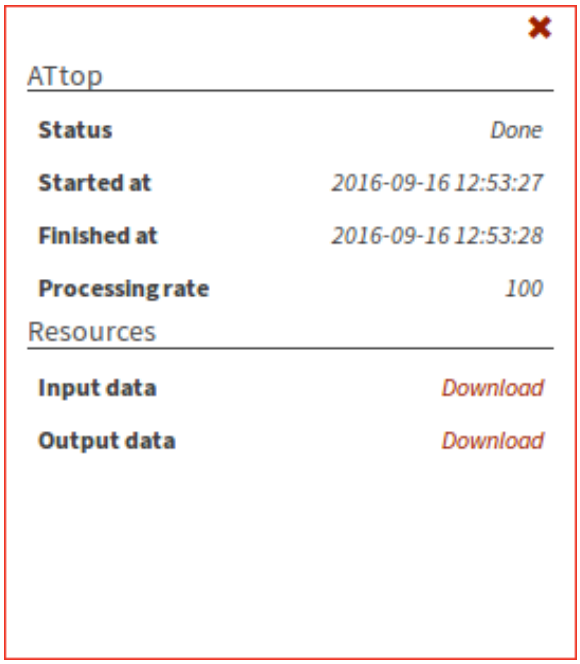

Figure 2.2.: Example of an executed task.

2.2. Methodology of the tool validation

In order to validate the TRE$_S$PASS integrated tools and applications, we took the sample input files provided by the TRE$_S$PASS partners and output files as a result. The following section provides these files, which are also available on the TRE$_S$PASS platform. We distinguish tools and applications, as explained in D6.4.3.

2.3. Overview of the test results

Application name	Version	Date of test	Results
Treemaker	1.0	2016-10-31 14:55:01	Success.
Software Checker	1.1.1.73	2016-10-21 15:14:12	Success.
Attack Tree Analyzer (ATA)	1.0	2016-10-31 15:00:50	Success.
Attack Pattern Library (APL)	-	2016-10-18 10:18:04	Success.
RawDataProcessing	-	2016-10-18 10:21:28	Success.
TRE$_S$PASS Model	-	2016-10-18 10:26:23	Success.
ATtop	1.0	2016-10-31 15:04:58	Success.

Table 2.1.: List of applications.

Tool name	Version	Date of test	Results
ADTool	2.2.1	2016-10-31	Success.
Attack Navigator Map (ANM)	0.34.6	2016-10-31	Success.
ArgueSecure offline	1.2.1	2016-10-31	Success.
ArgueSecure online	1.0	2016-10-31	Success.
Arg. spreadsheet	2.0	2016-10-31	Success.
TRICK Service	1.8.1-beta	2016-10-31	Success.
InterActor	1.0 Beta	2016-10-31	Success.
Architect	4.3.2	2016-10-31	Success.
Attack Tree Evaluator (ATE)	0.1	2016-10-31	Success.
e3tool	0.2.2 beta	2016-10-31	Success.
Attack-Defence Tree optimiser (ADTop)	1.0	2016-10-31	Success.

Table 2.2.: List of tools.

3. List of applications

The next sections provide examples of input and output files of the applications when executing toolchains. These applications are mainly applications executed with command lines, so no visible demonstration is really possible except through these input and output files.

3.1. Treemaker

3.1.1. Description

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 TRE$_S$PASS 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.

Where to find the application?
`https://trespass.itrust.lu/tools` in the "Analysis" category.

Current version of the application: 1.0.

3.1.2. Input

The input for Treemaker is a scenario that describes a model and a policy to invalidate. The policy can either be location-based (specifying that a certain asset is not allowed to reach a certain location) or action-based (specifying that a certain action is prohibited). Action-based policies usually take some asset as argument, to specify that the action is not permitted to be performed using these assets and having a certain role.

The two files model_IPTV.xml and scenario_IPTV.xml must be compressed together in a zip file like input.zip.

3.1.3. Output

Treemaker generates attack trees as XML files in the language for describing attack-defence trees used by the ADTool. Inner nodes in the generated trees have labels that indicate the actions performed by the sub-tree. Leaf nodes follow a grammar that enables their automatic parsing by tools such as the attack pattern library.

Files to be added when the program will be working.

3.2. Software Checker

3.2.1. Description

The Software Checker application in command line has been specifically developed in the context of the TRE$_S$PASS project so that it can be included in a tool chain.

Software Checker command line is part of the overall Software Checker tool, which is an applet collecting a list of programs (software packages) installed on a PC in order to check their potential security vulnerabilities.

Software Checker is a client/server application. It determines whether specific software versions contain any public vulnerability indexed by reliable databases such as CVE (http://nvd.nist.gov/). The CVE database is maintained up-to-date thanks to a daily synchronisation with the NVD (National Vulnerability Database) one. There are two clients:

- Windows Desktop;
- Java command line.

The Windows client checks for each authentication of a user whether one of the software packages installed on the machine (workstation, laptop, etc.) has vulnerabilities, whereas the client in command line uses information given by the user from the input sample file. This file can be completed using the following rule: every line represents a software. Its name and version have to be separated by a tabulation.

Where to find the application?
https://trespass.itrust.lu/tools in the "Misc" category.

Current version of the application: 1.1.1.73.

3.2.2. Input

SoftwareChecker_input.txt

3.2.3. Output

SoftwareChecker_output.txt

3.3. Attack Tree Analyzer (ATA)

3.3.1. Description

The attack tree computation tool, Attack Tree Analyzer (ATA), can be used to calculate optimal attack vector (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 the top ten most feasible attack vectors.

Where to find the application?
https://trespass.itrust.lu/tools in the "Analysis" category.

Current version of the application: 1.0.

3.3.2. Input

The input file provided on the TRE$_S$PASS platform are the following:

- Attack tree model with annotated leaves
- Attacker profile (optional)
- Genetic algorithm profile (optional, required only when analysis is done using genetic algorithms)

The tool expects an XML-based attack tree description. XML files corresponding to the corresponding XSD schema are considered valid and can be analysed.

approxtree_input.txt

3.3.3. Output

The output is typically the adversarial utility upper bound (security assessment is based on the value of adversarial utility upper bounds), and possibly some additional information. When the target enterprise is analysed using the genetic algorithms, the output file, apart from the utility estimation, contains the most profitable attack vector as well (if there is any).

approxtree_output.txt

3.4. Attack Pattern Library (APL)

3.4.1. Description

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.

3.4.2. Input

apl_input.xml

3.4.3. Output

apl_output.xml

3.5. RawDataProcessing

3.5.1. Description

The Raw Data Converter tool, RawDataProcessing, 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 TRE$_S$PASS input data.

3.5.2. Input

RawDataProcessing_input.xml

3.5.3. Output

RawDataProcessing_output.json

3.6. TRESPASS Model

3.6.1. Description

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.

3.6.2. Input

model_input.txt

3.6.3. Output

model_output.xml

3.7. Attack Tree Optimisation tool (ATtop)

3.7.1. Description

The Attack Tree Optimisation tool (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.

Where to find the application?
https://trespass.itrust.lu/tools in the "Analysis" category.

Current version of the application: 1.0.

3.7.2. Input

Inputs for the tool are provided in the ATtop's tool box:

- EnterBuilding.xml an attack tree model built with ADTool to describe an intruder trying to enter a building;
- StuxNet.at an attack tree model in the Galileo format to describe the StuxNet attack.

ATtop_input.xml

3.7.3. Output

ATtop_output.txt

3.7.4. Usage

In ATtop's interface, the input file can be uploaded by clicking on the "input" button and choosing the desired file from a local folder. Supported formats are ADTool (.xml) and Galileo (.at).

The required parameters are as follows:

- Max cost: the maximum cost allowed for an attack (INF stands for infinite);
- Max time: the maximum time (in arbitrary time units) an attack can take (INF stands for infinite);
- Query type: the type of query used to analyse the model. It can be one of the following:
  - Probability = probability of success;
  - Reachability = if success is possible, compute an attack vector;
  - ExpectedCost = expected cost of an attack;
  - OptimalSteps = if success is possible, compute the attack vector with the least number of steps;
  - OptimalTime = if success is possible, compute the attack vector which takes the least time.

After clicking the "Upload" button, the tool will be run and an output will be generated. Click on the "Download" link from the "Output data" line to obtain the output file(s). For all queries apart Probability, the output file will be a text file with the result in it. For example, a Reachability query will generate a sequence of steps that allow the attacker to succeed. A Probability query, apart from the probability interval [minimum, maximum] (which define a 95% confidence interval), generates also a cumulative probability distribution graph, which describes the probability of success at any given time point. The graph is provided both

as a .csv (comma-separated values) file with all the values and an already plotted graph as an image in .png format.

Examples:

1. Entering a building: use the EnterBuilding.xml file as input, and ask for a Reachability query with infinite bounds for both cost and time. Try different bounds for different results. A query of type OptimalSteps/Time will also work.
2. The StuxNet attack: use the StuxNet.at file as input, and ask for a Probability query with infinite cost bounds and 15 as maximum time. The minimum and maximum probabilities defining a 95% confidence interval at the given time point are provided, together with a cumulative probability distribution graph for all time points up to the given maximum. A query of type Reachability will also work.

4. List of tools

The next sections provide demonstration details on the tools as defined in D6.4.3, i.e. it gives more specific details about how they operate, and also output files as a result.

4.1. Attack–Defense Tree Tool (ADTool)

4.1.1. Description

The Attack–Defense Tree Tool (ADTool) (Kordy, Kordy, Mauw, & Schweitzer, 2013a, 2013b) allows users to model and display attack–defense scenarios, through the use of attack–defense trees (ADTrees) (Kordy, Mauw, Radomirović, & Schweitzer, 2012) or an alternative term-based representation of ADTrees called attack–defense terms (ADTerms). It supports the methodology developed within the ATREES project (Kordy et al., 2009–2012). Since attack trees (Salter, Saydjari, Schneier, & Wallner, 1998; Mauw & Oostdijk, 2006), protection trees (Edge, Dalton II, Raines, & Mills, 2006), and defense trees (Bistarelli, Fioravanti, & Peretti, 2006) are formally instances of attack–defense trees, the ADTool can also be employed to automate and facilitate the usage of all aforementioned formalisms. Furthermore, the ADTool allows to perform quantitative analyses on ADTrees/ADTerms. This means that a user is able to answer questions such as: What are the costs of an attack, what is the minimal skill level required for the attacker, how long does it take to implement all necessary defenses or who is the winner of the considered attack–defense scenario, and many others.

In short:

- The ADTool allows the user to model attack–defense scenarios using ADTrees and ADTerms;
- The ADTool allows the user to perform quantitative analyses on ADTrees/ADTerms.

In detail:

- Graphical and user-friendliness features:

  - Various tree display methods are available. For example it is possible to collapse and expand trees at all nodes, to move and center the tree in all nodes, and to zoom into and out of the tree. These functions allow a user to model large, real-life scenarios;
  - A user can export ADTrees to image or LaTeXfiles;
  - Input and editing of ADTrees and ADTerms can be done with the keyboard as well as the mouse;
  - Various themes and layouts enhance the user experience.
- Quantitative and computational features:
  - The tool automatically converts back and forth between ADTrees and ADTerms. Among others, this provides a tool to learn one of the syntaxes, provided the other one is known;
  - The tool possesses full ADTerm syntax handling including syntax error highlighting;
  - A large number of commonly used attribute domains have been specified and are universally usable;
  - A user can save/load attack–defense scenarios with attribute domains and attribute values.

Where to find the tool?
https://trespass.itrust.lu/tools in the "Visualisation" category.

Current version of the tool: ADTool 2.2.1 (October 20, 2016 - 79c77e).

4.1.2. Connection

When you click on the "Run" application link in the TRESPASS platform, a file named "ADTool.jnlp" is downloaded (accept the security message). Double-click on it to launch the application (accept the below execution warning message).

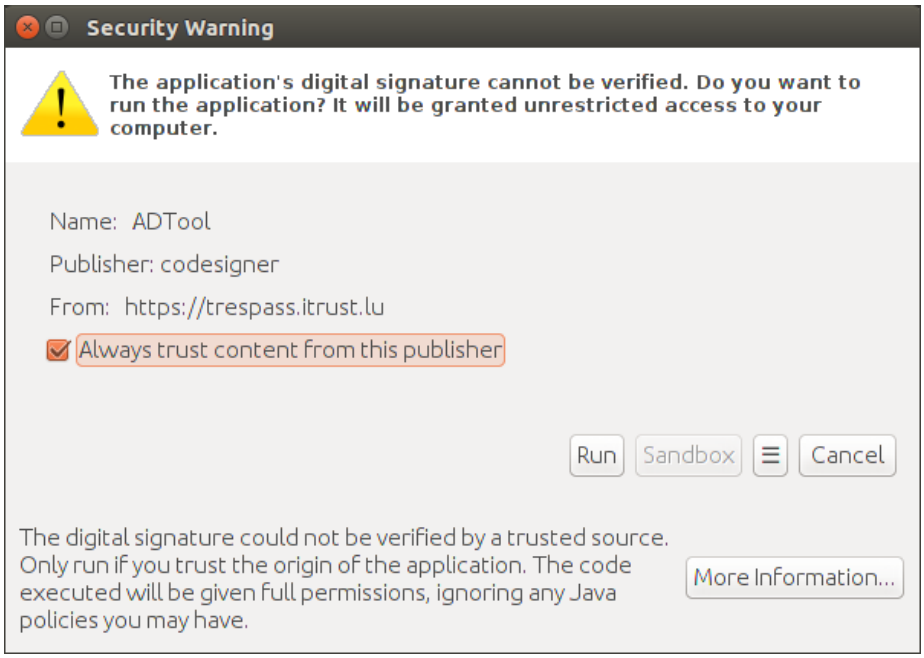

Figure 4.1.: Example of execution warning message.

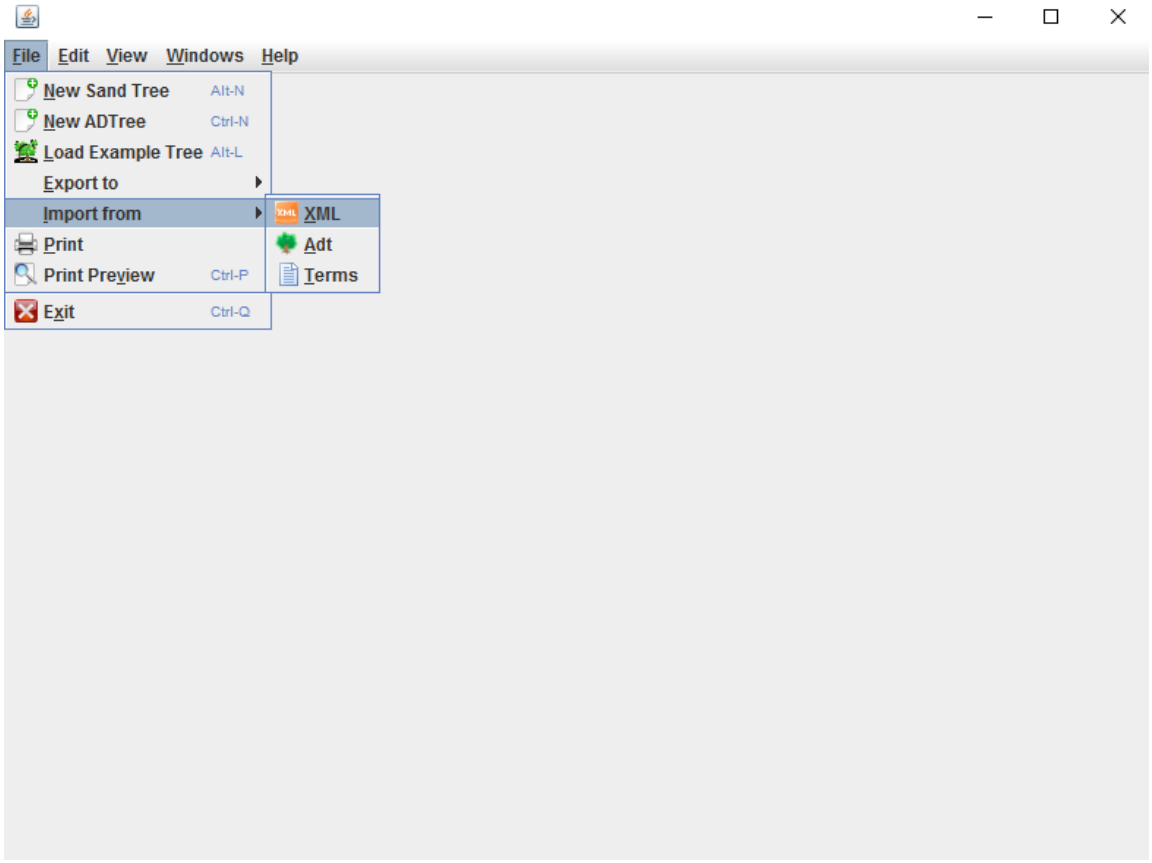

Figure 4.2.: ADTool.

4.1.3. Usage

See the user manual for more information at the following link:
http://satoss.uni.lu/members/piotr/adtool/manual.pdf

4.2. Attack Navigator Map (ANM)

4.2.1. Description

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 and a set of attacker profiles.

The Attack Navigator Map represents the system cartographically, displaying connections between the elements as potential steps that an attacker could take. These steps are annotated with relevant variables such as difficulty and cost.

The attacker profile collects relevant characteristics of an attacker, such as skills, resources, motivations / goals, and initial access. By combining a map and attacker profile, the system will calculate routes for the attacker across the map that provides utility to the attacker.

Typically, this will involve gaining access to certain assets and compromising their confidentiality, integrity or availability, which may cause damage to the organisation. The routes with the highest utility for the attacker constitute the highest risk with respect to the selected attacker profile.

Various tools analyse the various routes, and the results are visualised in a dashboard for inspection. On the basis of the outcomes, a user can implement countermeasures and rerun the analysis, until satisfied.

Where to find the tool?
https://trespass.itrust.lu/attack-navigator-map/.

Current version of the tool: 0.34.6.

4.2.2. Connection

When you click on the "Run" application link in the TRE$_S$PASS platform, you arrive to the Attack Navigator Map (ANM) web application home page.

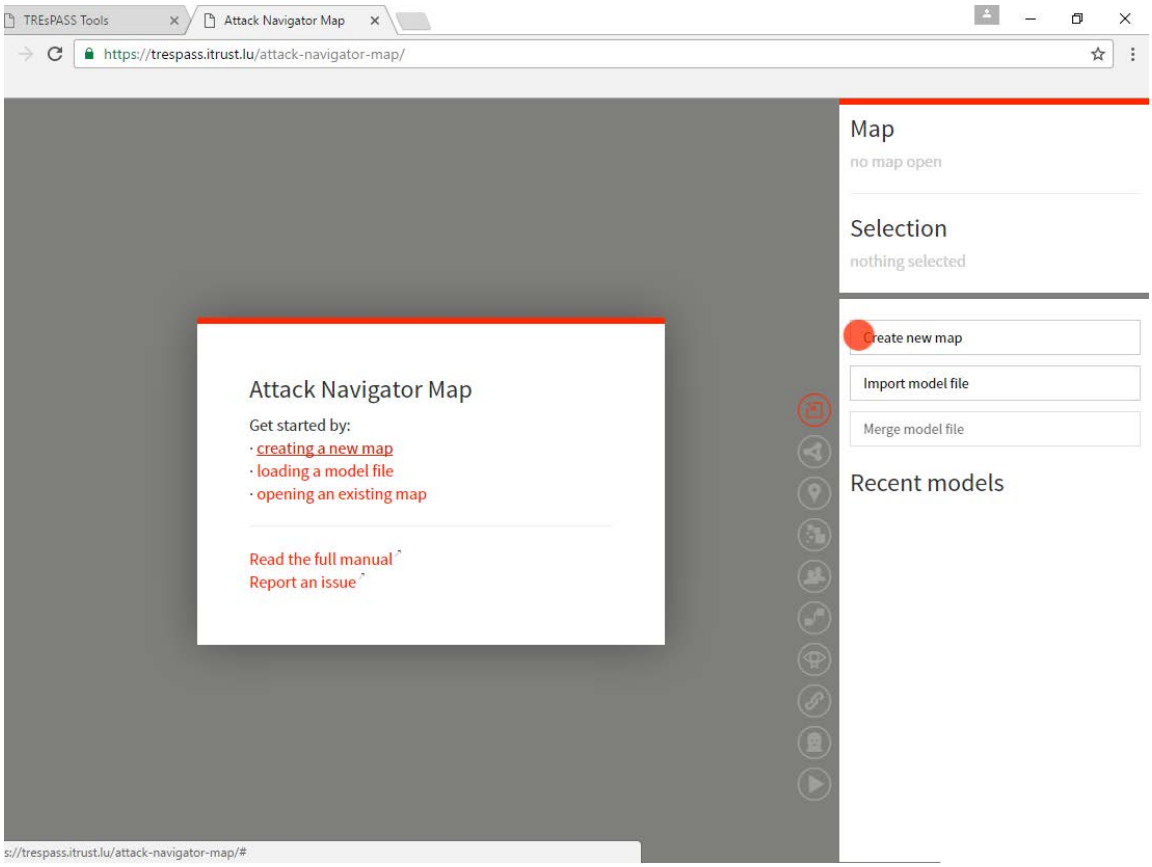

Figure 4.3.: ANM home page.

4.2.3. Usage

See the ANM manual for more information at the following link:
https://docs.google.com/document/d/1Qp8nJgdvDespq1Q5zQcAT1SSTKK23m2XKMUKKmoKYQU/edit?usp=sharing

4.3. ArgueSecure online

4.3.1. Description

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:

- An Excel-based version;
- A Java version;

- An online version.

The online version of ArgueSecure is designed to be usable with a projector, and in addition allows stakeholders and experts to engage in a risk assessmentin real-time without being in the same room and even without being available the same time.

All three tools share a similar way of encoding risk arguments, but have slightly different functionality, described on the GitHub pages (for the online and Java version) and in the spreadsheet (for the Excel-based version).

The ArgueSecure tools are based on research into argumentation-based risk Assessment tool based on previous research by Ionita *et al.*: http://eprints.eemcs.utwente.nl/25041/01/Argumentations-support_for_Security_Requirements.pdf.

Where to find the tool?
https://trespass.itrust.lu/tools in the "Data collection" category.

Current version of the tool: 1.0. The following web page lists links to the latest versions of each tool: https://danionita.github.io/ArgueSecure/. For the online version, users have to do a Git pull to ensure that they have the latest version.

4.3.2. Usage

Users need to create an account, log in, then fill in with the claims being made with respect to risks, attacks and defences, encoded using the structure provided.

4.3.2.1. Creating and opening assessments

After logging in, you will be presented with a list of available risk assessments. These are either assessments that you created or public assessments created by other users.

- Open existing assessment: simply click on the name (or the "Open assessment" button) of any of the available assessments to open;

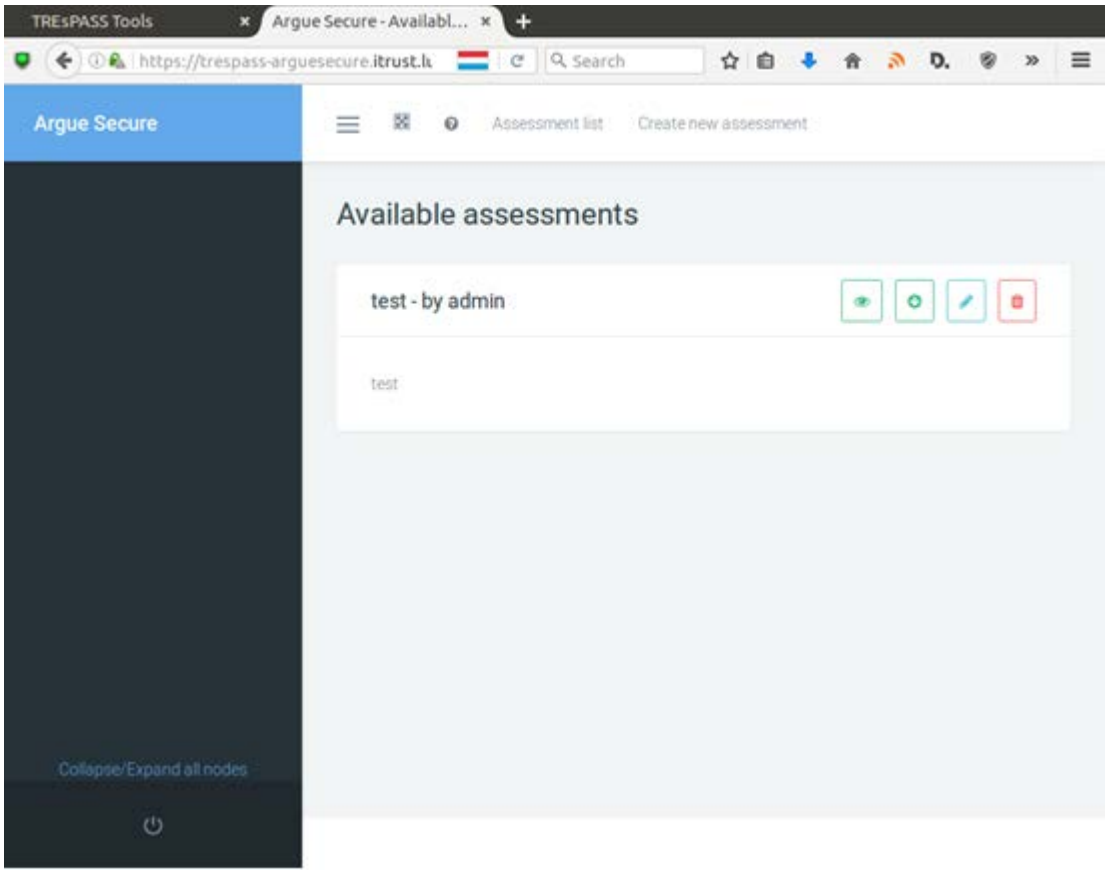

Figure 4.4.: Available assessments.

- Create new assessment: you may also create a new assessment by clicking the “Create new assessment” button on the top menu;

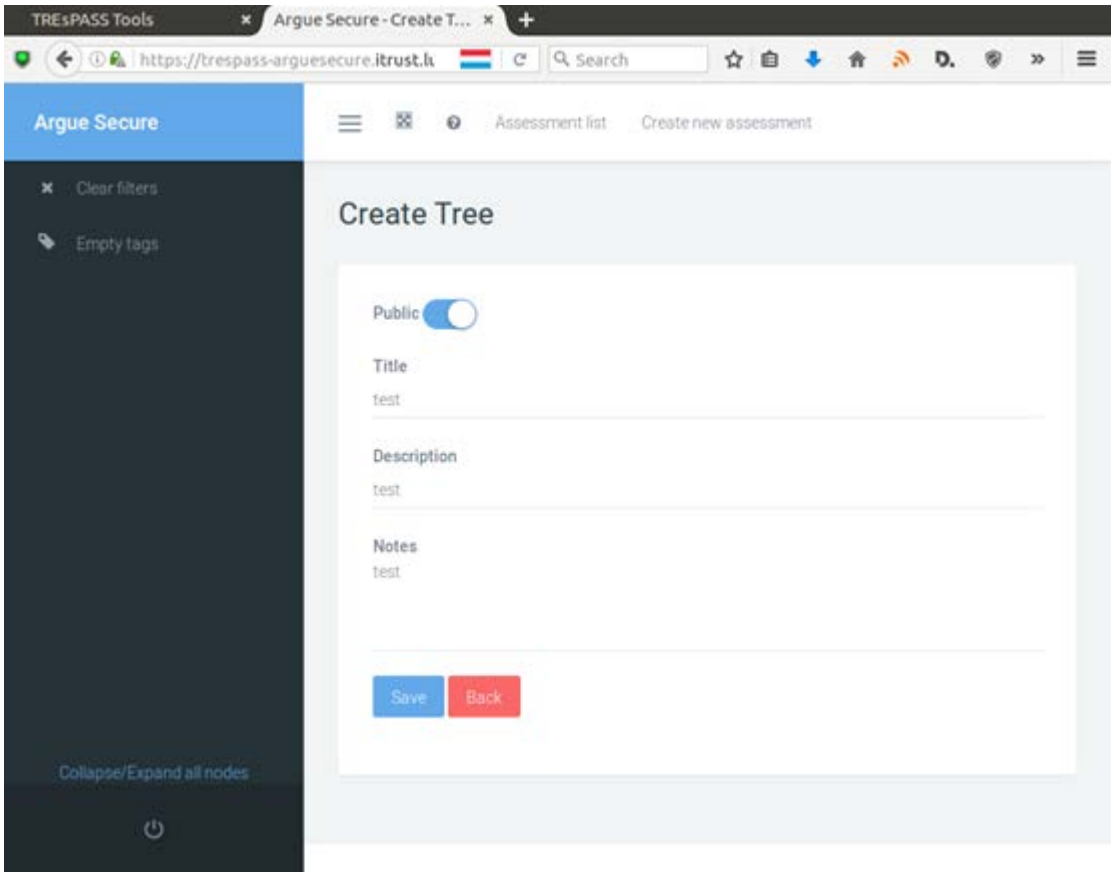

Figure 4.5.: Create new assessment.

- Edit an assessment: the "edit assessment" button allows you to change the name and description of an assessment, as well as making its sharing settings. It is only available for assessments created by you;

- Delete assessment: the "delete assessment" button only appears if the assessment was created by you;

- Export assessment: the "export assessment" button produces an indented list of all the content of an assessment.

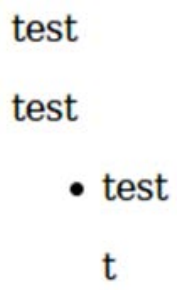

Figure 4.6.: Example of exported assessment.

4.3.2.2. Contributing to an assessment

Once you have opened an assessment (or created a new one), you can start visualizing, adding and modifying content:

- Toggle node description: to show/hide the textual description attached to a node, click its "show description" button. This button only appears if the node has a description.
- Toggle node notes: to show/hide the notes attached to a node, click its "show notes" button. This button only appears if the node has a note.
- Create new node: to create a new node, click the "Show actions" button of the node under which you want to insert your new node and then the "Add" button
- Edit a node: to edit the name, description, tags or other attributes of a node, click its "Show actions" button and then the "Edit" button. You may also quickly edit the name of the node by left-clicking on it.
- Delete a node: to remove a node description, tags or other attributes of a node, click its "Show actions" button and then the "Edit" button
- Collapse/expand subtree: to collapse or expand the sub-nodes of a certain node, double click the node. If the node is expansible/collapsible, a +/- button will appear in the top-right corner of the node.

The online tool has installation and configuration instructions in the README.md file visible on the project's main GitHub page https://github.com/hitandyrun/arguesecure-online and usage instructions are present in its built-in help page.

Pre-configured virtual machines are available at https://surfdrive.surf.nl/files/index.php/s/0C1tM7suCRGJIN0.

ArgueSecure online and offline tools are documented in "ArgueSecure: Out-of-the-box risk assessment" (Ionita, Kegel, Baltuta, & Wieringa, 2016).

4.4. ArgueSecure offline

4.4.1. Description

This is the Java version of ArgeSecure, which is intended to be used during dedicated security requirements elicitation sessions. It is designed to be usable with a projector.

ArgueSecure allows you to build and maintain a list of risks with the following structure:

- Category: <A category of risks>
  - R1: <a risk>
    - (sword) C1: <Claim made by an attacker about the existence of an attack path>
      - A A1.1: <An assumption of the claim>
      - A A1.2: <Another assumption of the claim>
    - (shield) C2: <Claim made by a defender, that partly or completely defeats the attacker's claim by pointing out that an attacker's assumption is probably, or certainly, false>
      - A A2.1: <An Assumption of the defender's claim, e.g. about a mitigation that already exists or that will be implemented.>
    - (sword) C3: <Renewed claim of the attacker that bypasses the defender's argument>
      - A A2.1: <An assumption of this renewed claim>
  - Etc.
  - R2: etc.
- Category: etc.

Defender's arguments can refer to components or architectural decisions that reduce a risk, and to decisions to transfer some risk to a third party (e.g. to an insurance company or to a customer). If a risk ends with all attacker's claims partly or completely defeated, then all attack paths claimed by attackers in this argument have been partly or completely mitigated, or transferred to third parties. Risks that are undefeated are accepted by the defender. The tool allows the production of lists of mitigations per risk, and of risks per mitigation. It provides a memory of the reasons why mitigations have been introduced, and which risks have been considered for mitigation.

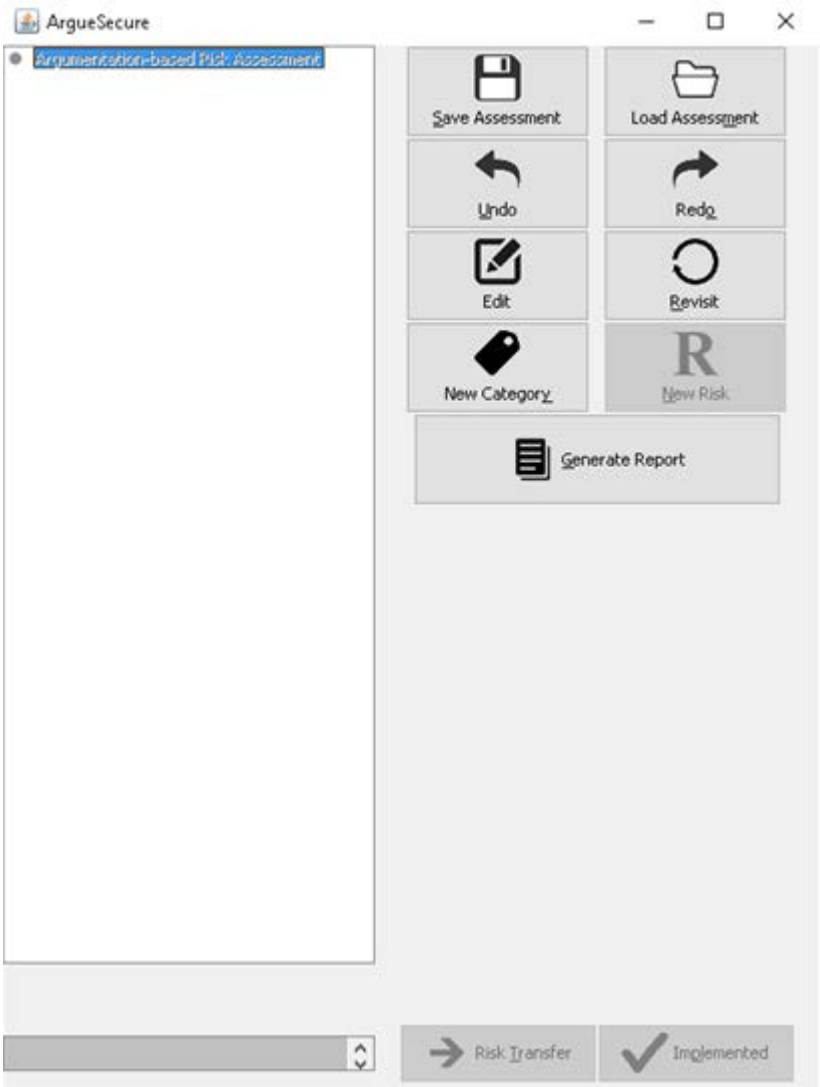

Figure 4.7.: ArgueSecure offline.

Where to find the tool?
https://trespass.itrust.lu/tools in the "Data collection" category.

Also here https://danionita.github.io/ArgueSecure/.

The offline version's code is hosted at https://github.com/danionita/ArgueSecure, and binaries at https://github.com/danionita/ArgueSecure/releases/download/v1.2.1/ArgueSecure.jar.

Current version of the tool: 1.2.1. For the offline version, the latest release is available at https://github.com/danionita/ArgueSecure/releases/latest.

4.4.2. Usage

Conducting an argument-based risk analysis requires little preparation. Any number of stakeholders, domain experts and/or security experts can participate, but should be split up in two teams: "Attackers" and "Defenders". The method assumes the participants possess pre-existing knowledge of the Target of Assessment. Ideally, but not mandatory, some sort of system model or diagram should be agreed upon by the participants. The preferred process to use the tool is as follows:

1. Create a new risk and give it a name.
2. Create a new risk under this category, and provide a brief name/description of it.
3. Each risk starts with an attacker argument, describing an attack path or refining the risk. Each argument consists of a claim, supported by one or more assumptions. Use CTRL+SPACE to switch between entering claims and assumptions.
4. Each attacker argument may be countered by a defender argument, describing a mitigation, reduction or transfer of the risk.
5. This back-and-forth rhetoric can continue until:
   - The attacker team is unable or unwilling to counter the last defender argument. This means the risk has been mitigated sufficiently for this attacker.
   - The defender team is unable or unwilling to counter the last attacker argument. This means the (residual) risk has been accepted.
6. If other risks can be identified under this category, go back to step 2 and create a new risk.
7. If a new risk category can be identified, go back to step 1.
8. At any time during the assessment, defender arguments can be marked as "Implemented" (if they describe existing risk countermeasures) and/or "Transfer" (if they describe a risk transfer).

4.4.2.1. Colour codes

- Black: the colour of all elements except claims;
- Green/red: only applies to claims. Claims start out as green and turn red once defeated. Then, turn green again once their counter-arguments have been defeated and so on.

4.4.2.2. Tips & Tricks

- When to start a new risk?
  - As soon as a new attack vector is identified (even if this attack vector intuitively corresponds to the same risk and/or compromises the same asset), it is recommended to specify it as part of a new risk to prevent long rounds.
- Contents of "Attacker" arguments:
  - Claims are attacks that are possible;
  - If everybody agrees, no assumptions are needed. If anyone believes it's impossible, he has to explain why (in which circumstances);
  - The negations of these reasons are assumptions.
- Contents of "Defender" arguments:
  - Claims are (parts of) attacks which are impossible;
  - If everybody agrees, no assumptions are needed. If anyone believes it's still possible, he has to explain why (in which circumstances);
  - The negations of these reasons are assumptions.

4.5. ArgueSecure spreadsheets

4.5.1. Description

This is the lightweight Excel-based version of ArgueSecure.

Where to find the tool?
https://trespass.itrust.lu/tools in the "Data collection" category.

The spreadsheets are downloadable from the TRESPASS platform or here:
https://surfdrive.surf.nl/files/index.php/s/lltTF8Vms588amz.

Current version of the tool: 2.0.

4.5.2. Usage

Before starting with the tool, users need to fill in with the claims being made with respect to risks, attacks and defences, encoded using the structure provided:

- Users need to fill in names and IDs of assets in columns U, V (optional);
- For each argument:
  - Split up argument into Claim + Assumptions + Facts and write these down in the respective (txt) cells;
  - # cells will be filled in automatically;
  - For counter-arguments, also add the # of the argument part it is invalidating in the Rebuts cell (optional).

Orange boxes are mandatory, yellow boxes are optional, grey boxes are automatically filled in.

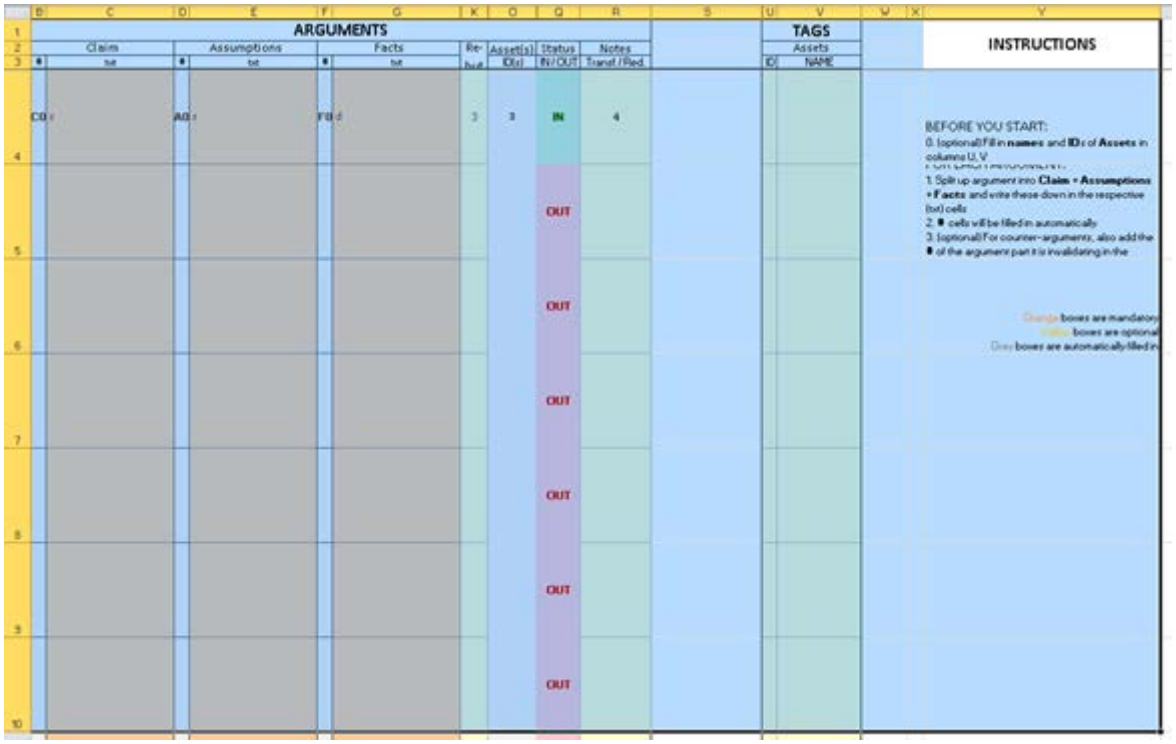

Figure 4.8.: ArgueSecure spreadsheets.

The spreadsheet contains usage instructions within the spreadsheet itself and here is also a text file with some extra hints on differentiating between claims and assumptions: ArgueSecure spreadsheets.txt

ArgueSecure spreadsheets is additionally documented in "Argumentation-Based Security Requirements Elicitation: The Next Round" (Ionita, Bullee, & Wieringa, 2014).

4.6. TRICK Service

4.6.1. Description

TRICK Service is an easy to use web application designed to conduct risk assessments according to ISO/IEC 27005 or CSSF 12/544. A wide variety of features such as multi-user support, quantitative analysis of risk scenarios, maturity assessment, access controls, import/export, risk analysis versioning, definition of risk profiles, embedding of own security control catalogues or international standards lead to an efficient risk management.

Where to find the tool? https://trespass-trickservice.itrust.lu.

Current version of the tool: 1.8.1-beta.

4.6.2. Connection

When you click on the "Run" application link in the TRESPASS platform, you arrive to the TRICK Service web application homepage. Enter the credentials you have been provided by itrust consulting to start with a new risk analysis.

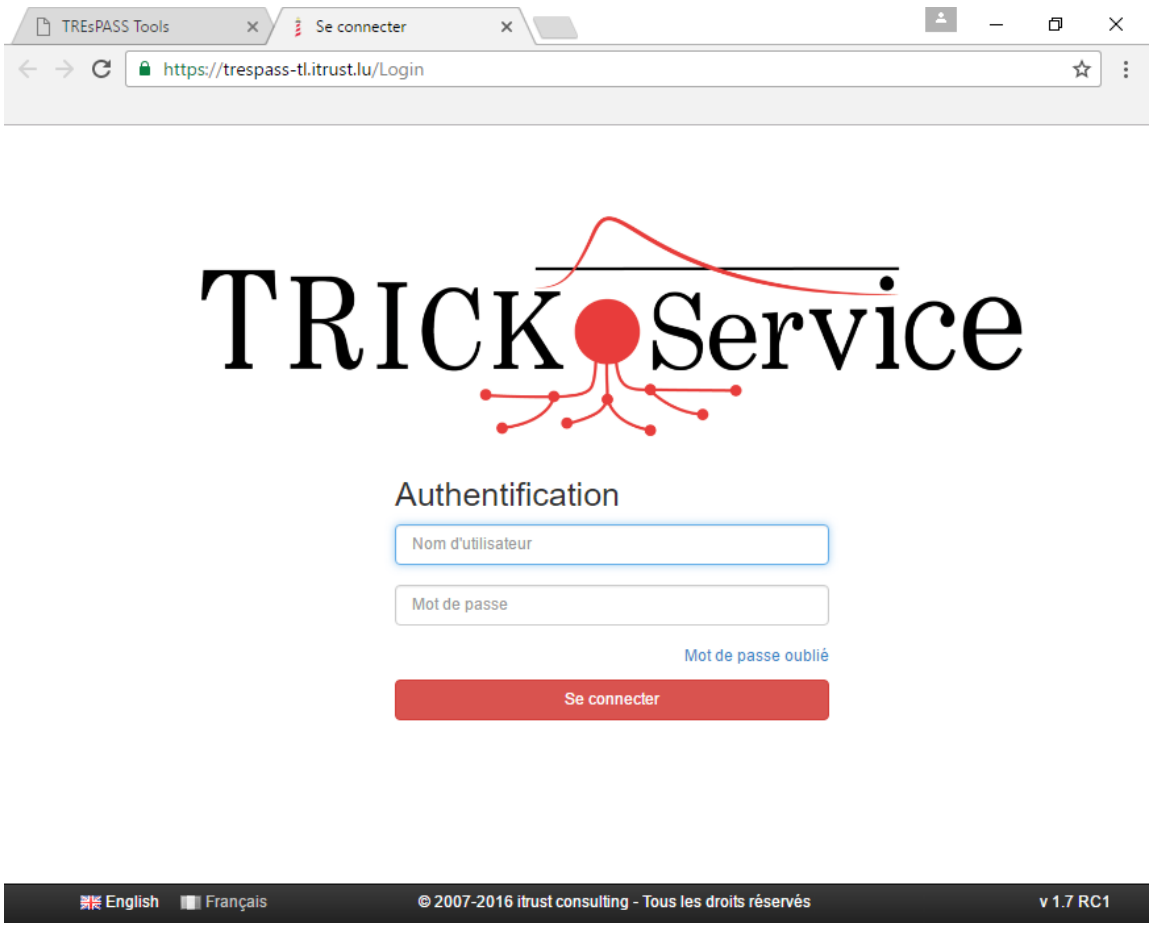

Figure 4.9.: TRICK Service home page.

4.6.3. Usage

See the TRICK Service user guide attached here for more information:

TRICK Service user guide.

4.7. InterActor

4.7.1. Description

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.

Where to find the tool?
https://trespass.itrust.lu/tools in the "Data collection" category.

Current version of the tool: 1.0 Beta.

4.7.2. Connection

The first time users access the tool, they need to enter a new username and password by clicking on the "Register" menu at the top of the welcome page.

Figure 4.10.: Register to InterActor.

In order to connect, users then need to click on the Login link, enter the previous username and password, and click on Sign In.

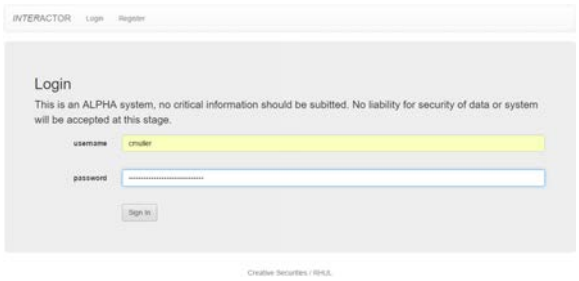

Figure 4.11.: Login to InterActor.

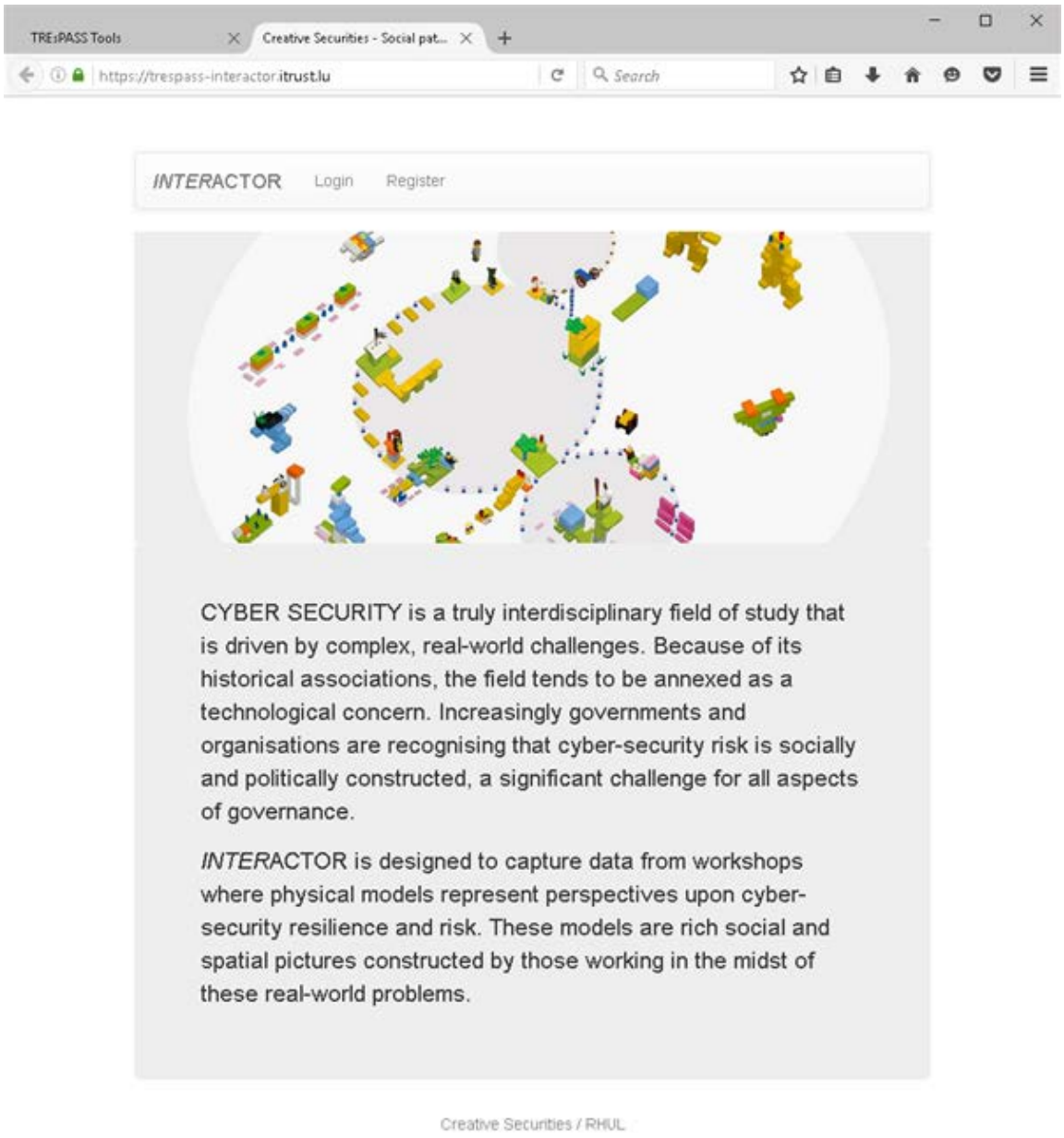

Figure 4.12.: Welcome page of InterActor.

4.7.3. Usage

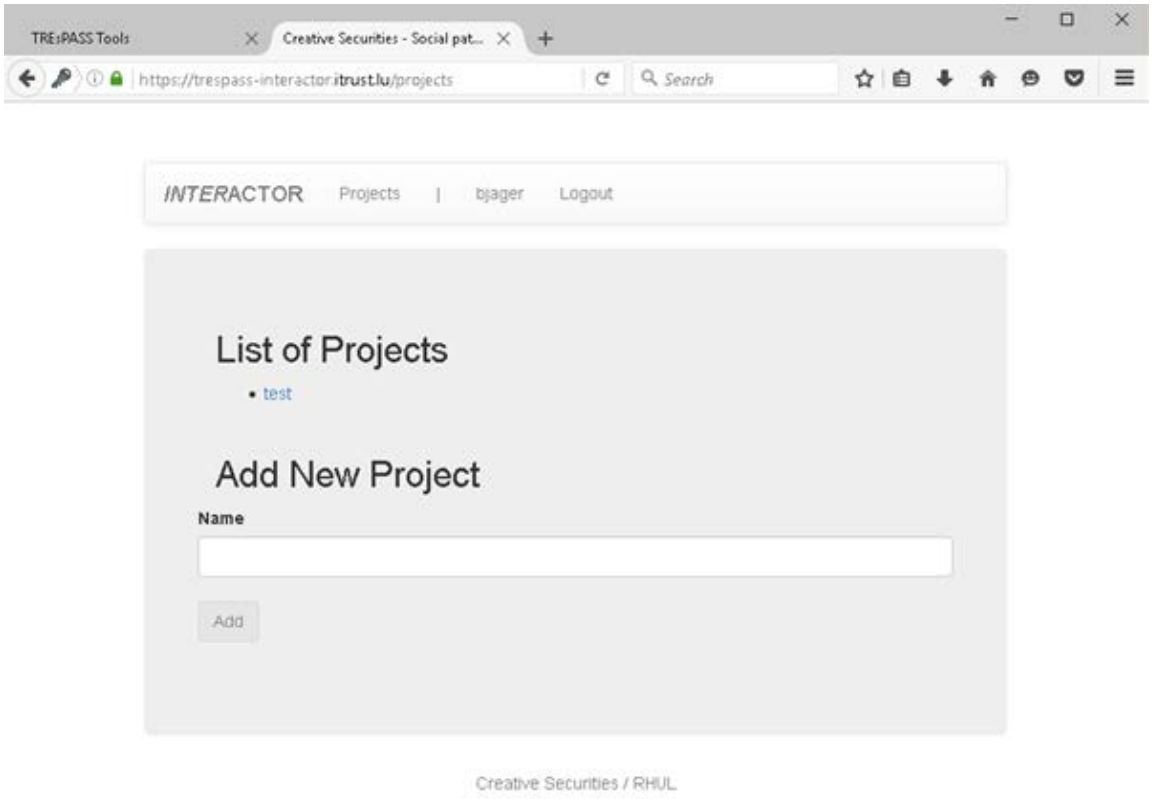

Figure 4.13.: List of projects.

When connected, users can use the "Projects" link to see the list of current projects, or create a new project using the "Add New Project" form: enter a name, then click on "Add" to add a new project.

4.7.3.1. Editing a project

Click on the name of the project to edit it. It displays a list of workshops, and a form to add a new workshop. It also displays a form to add a demo workshop.

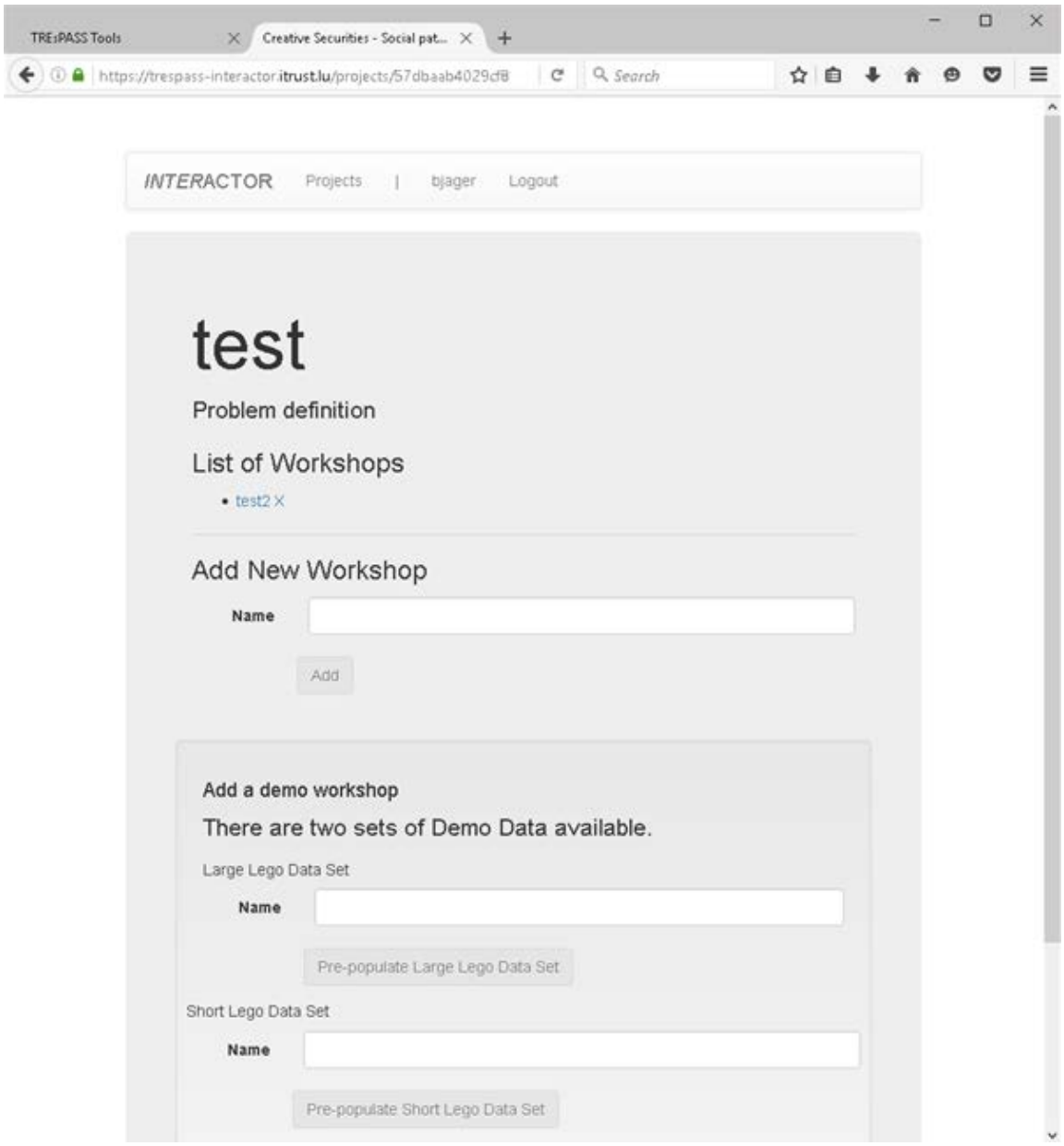

Figure 4.14.: Edit a project.

4.7.3.2. Opening a workshop

Clicking on the name of a workshop opens it. Following picture is a demo workshop:

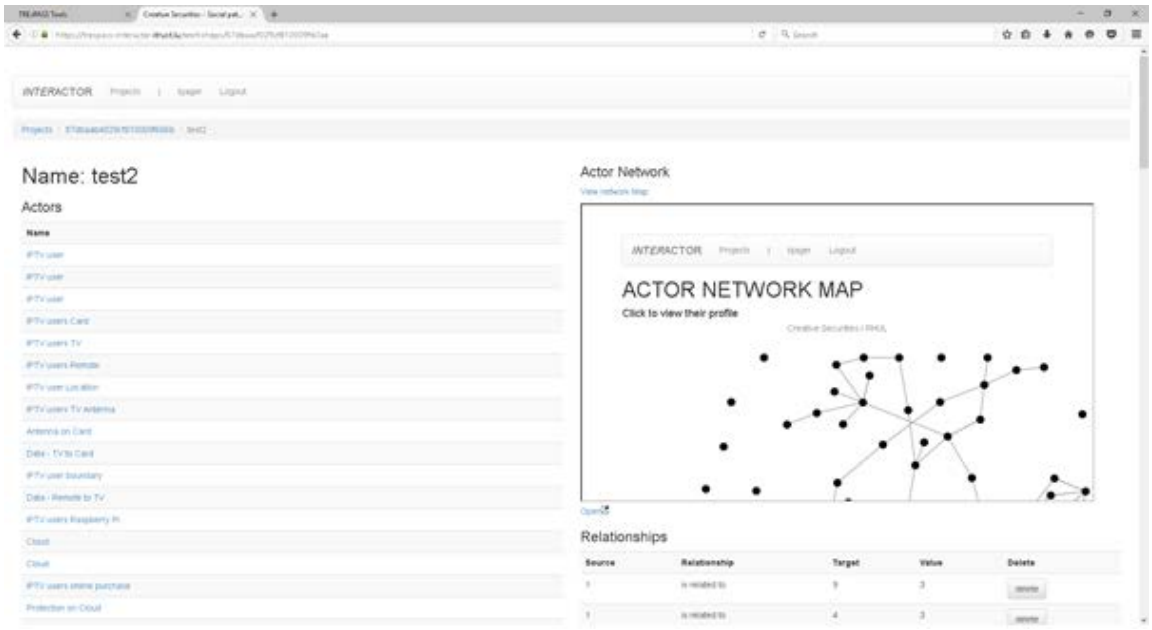

Figure 4.15.: Demo workshop.

The sample data file loads on request by the user.

Users can check that the corresponding output is correct by visually cross-referencing with the physical models and the annotations made by the group within the tool and in personal notes and drawings.

The documentation of InterActor will be found on the Help page of the tool. It is planned that a demo video will be embedded there in addition to textual guidance. Finally, in time the tool will be available on other servers with a new domain name, but we are unable to give details about this at the moment.

4.8. Architect

4.8.1. Description

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).

Where to find the tool?
https://trespass.itrust.lu/tools in the "Modelling" category.

Current version of the tool: 4.3.2.

4.8.2. Connection

When you click on the "Run" application link in the TRESPASS platform, a file named "cpub-Architect-BiZZdesign_Hosti-CmsRdsh.rdp" is downloaded (accept the security message).

Double-click on it (accept the execution warning message) and enter credentials you have been provided by BiZZdesign in order to start creating a new model.

Login: BIZZDESIGNHOST\<username>, domain: BIZZDESIGNHOST

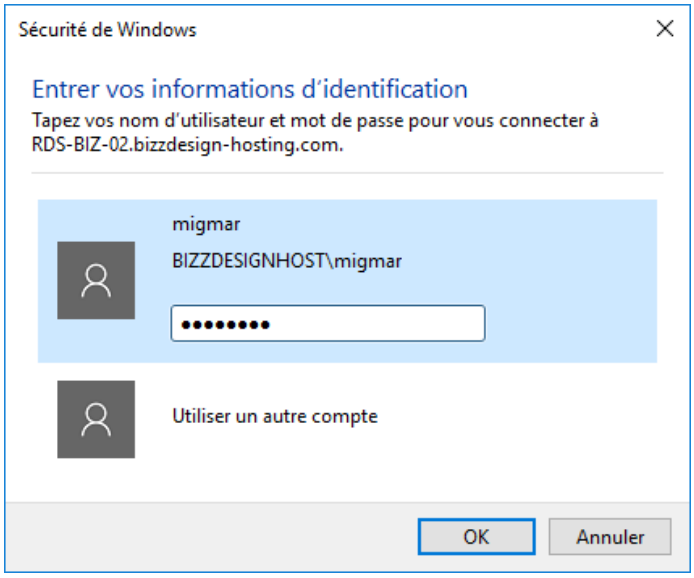

Figure 4.16.: Connection to Architect.

The application opens remote access to server and opens Architect.

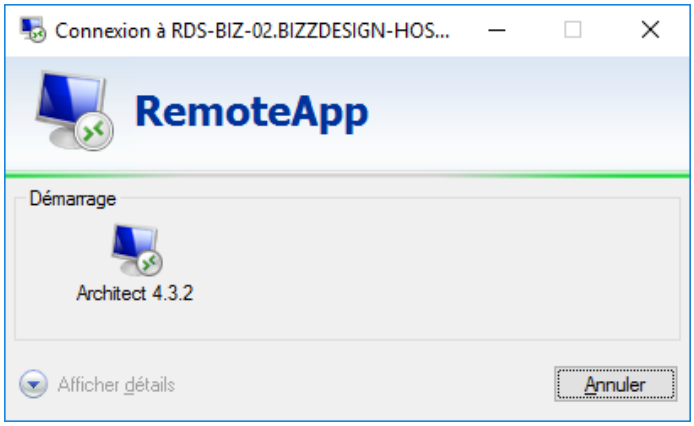

Figure 4.17.: Open remote access to server.

Figure 4.18.: Opening Architect.

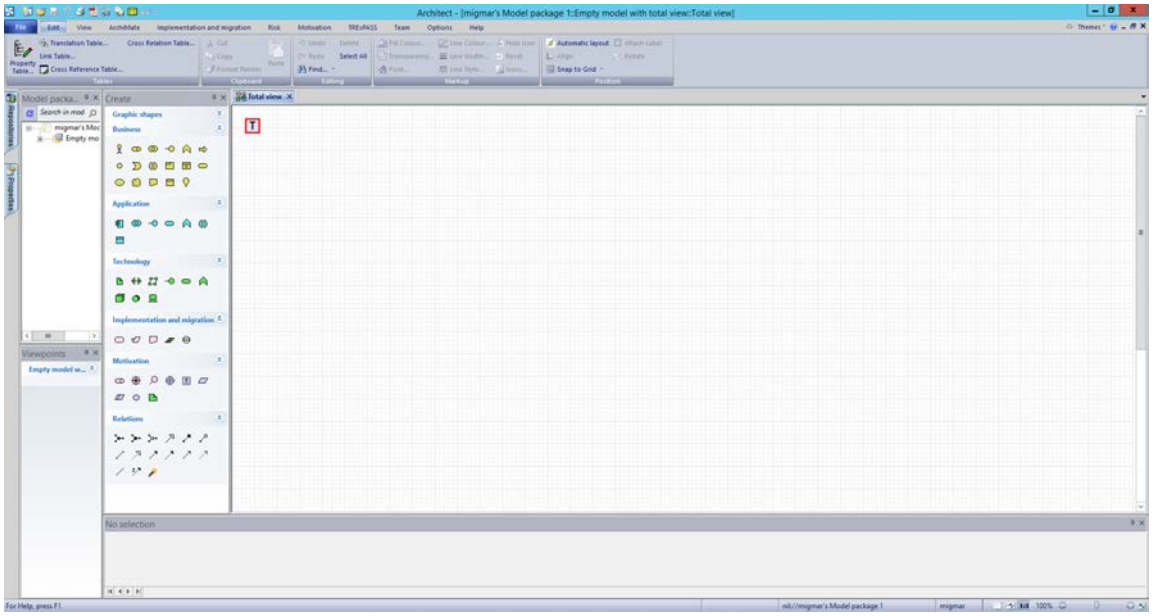

Figure 4.19.: Welcome page of Architect.

4.8.3. Usage

For using the Architect tool, a user guide is available here: Architect user guide.

4.9. Attack Tree Evaluator (ATE)

4.9.1. Description

The Attack Tree Evaluator (ATE) 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 the ATE allow the defender for example to identify the maximum probability for a successfull attack, given an attacker budget.

Where to find the tool?
https://trespass.itrust.lu/tools in the "Analysis" category.

Current version of the tool: 0.1.

4.9.2. Input

The tool takes as input an attack tree and the values for the basic actions such as probability of success and eventually their cost. The input attack tree for the tool should be in XML format that can be generated from the socio-technical model by other tools in the TRESPASS project, i.e., the Treemaker. The values for the basic actions are expected to be in the XML source.

4.9.3. Output

Depending on the values provided for the basic actions, the tool can compute different outputs. When only the probability values are provided for the basic actions, the tool computes the maximum success probability of an attack. When the probability and cost values are provided for basic actions, the tool computes the Pareto set of attacks with maximum success probability and minimum cost. In both cases, together with the solution the tool provides the set of corresponding basic actions. The tool can also compute the minimum cost of an attack, if the probability values are either 0, meaning the basic action is not performed, or 1, meaning the basic action is performed.

4.9.4. Connection

When you click on the "Run" application link in the TRESPASS platform, a file named "AttackTreeEvaluator.jnlp" is downloaded (accept the security message). Double-click on it (accept the execution warning message) to launch the ATree Evaluator (ATE).

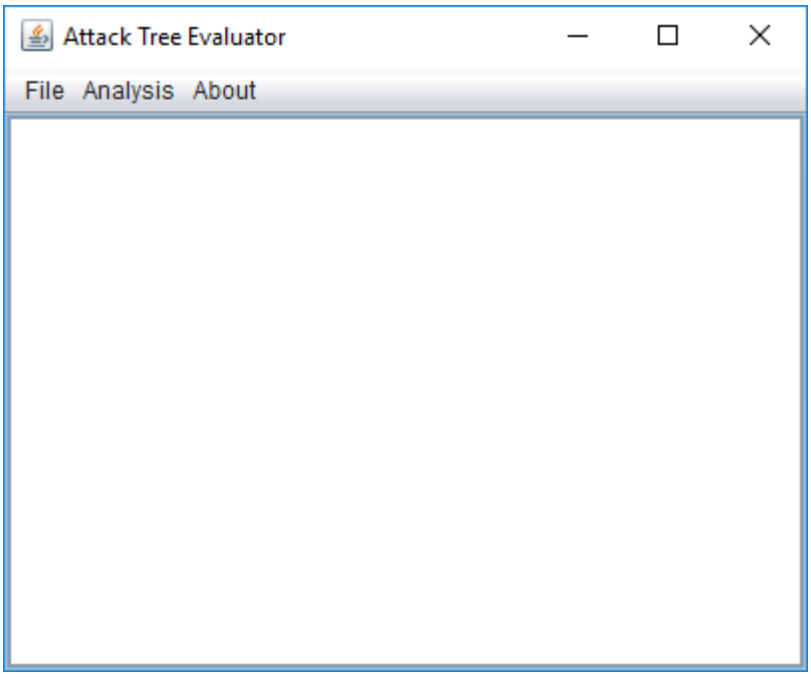

Figure 4.20.: Attack Tree Evaluator.

4.9.5. Usage

The ATE is developed in Java and distributed as a jar archive, therefore a Java Virtual Machine (JVM, the Java runtime environment) is required to run the program, and the JVM version should be greater or equal to 1.7.0.

The ATE requires as input trees in the XML format defined by the TRE$_S$PASS project.

To run the tool, from a shell or command prompt execute:

```
java -jar ATE.jar <path-to-tree>.xml <pareto|probability>
```

where the parameter 'pareto' launches the analysis maximising the probability of success and minimising the cost, and the parameter 'probability' only maximises the probability of success. The values of probability and cost to basic actions are expected to be in the XML source. In this implementation, the probability can be specified as a rational number from 0 to 1 or with a label in L, M, H, V which is then assigned a value in 0.1, 0.4, 0.7, 0.9, respectively. In case a numeric value is provided, the system settings should support the English separators.

4.10. e3tool

4.10.1. Description

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 visualize fraudulent variations of a given value model.

e3tools is a merger between two existing projects:

- The e3editor tool with which users can construct and evaluate value models of networked business models:
  - Represent networked business models in terms of end users and enterprises, as well as the things of economic value they exchange with each other;
  - Assign economic value to the things exchanged, set pricing models, the number of customer needs, the actors involved and required investments;
  - Calculate the profitability of a networked business model for all actors involved, evaluate changes in participating actors, and increase of prices.
- The e3fraud tool with which users can conduct fraud assessment of a networked business model:
  - Automatically generate deviations from a given value model, in terms of (1) transactions that might not occur as agreed, (2) transactions that were not expected to occur and (3) collusion, where two or more actors thought to be independent act together against the interests of the provider;
  - Sort and filter these fraud scenarios based on gain for potential fraudsters or impact for the service/product provider;
  - Visualise the financial effects of fraud scenarios across a given range of occurrence rates.

By merging these two tools, users can now also manually construct (and analyse) e3fraud models.

Where to find the tool?
https://trespass.itrust.lu/tools in the "Analysis" category.

Current version of the tool: 0.2.2 beta. See latest releases at https://github.com/danionita/e3tools/releases.

4.10.2. Input

The e3tool is a standalone tool so it does not take any input.

4.10.3. Output

The e3tool can produce profitability charts from "normal" and fraud value models as well generate ranked lists of fraud scenarios and associated models.

4.10.4. Connection

When you click on the "Run" application link in the TRE$_S$PASS platform, a file named "e3tools.jnlp" is downloaded (accept the security message). Double-click on it in order to start using e3tool.

4.10.5. Usage

4.10.5.1. Editing a model

You can drag and drop elements from left panel to the right panel to edit a value model.

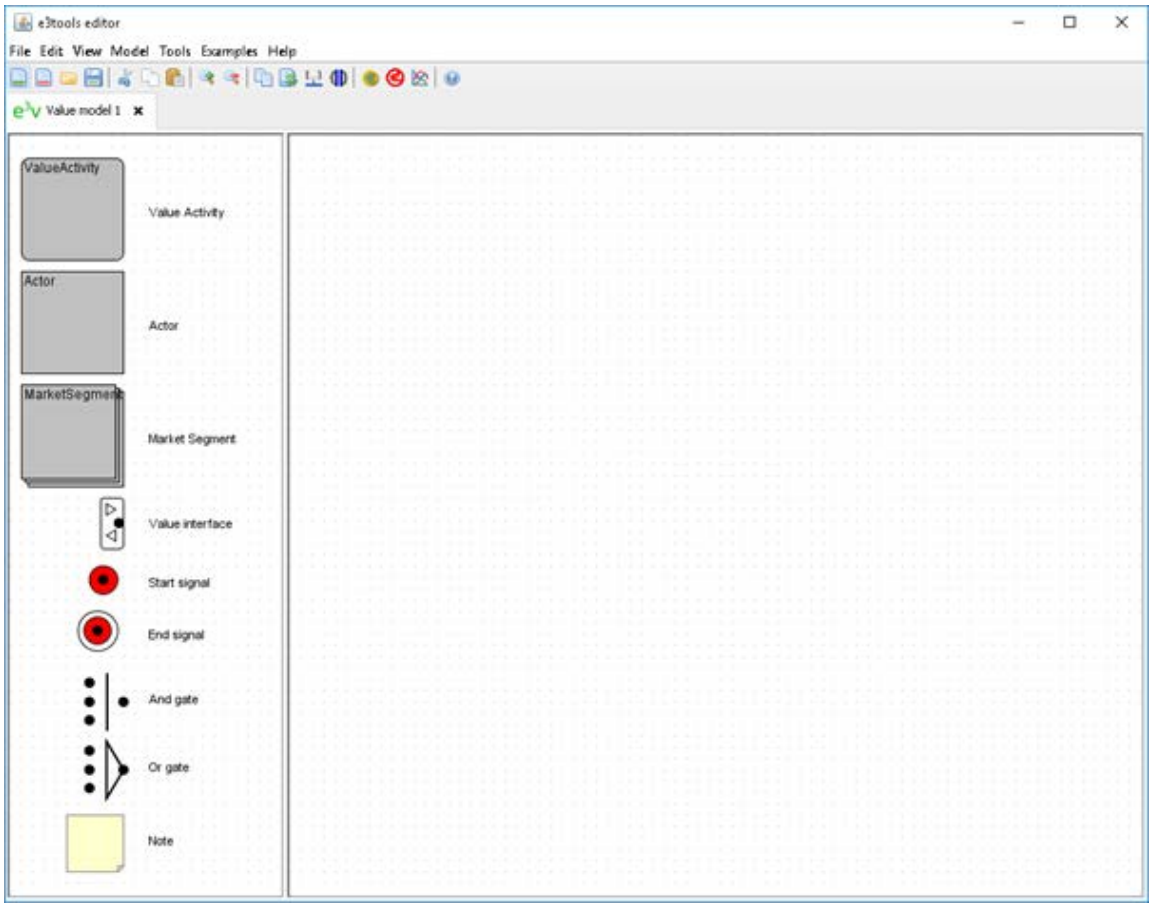

Figure 4.21.: e3tool editor.

Next screenshot shows an example of a value model.

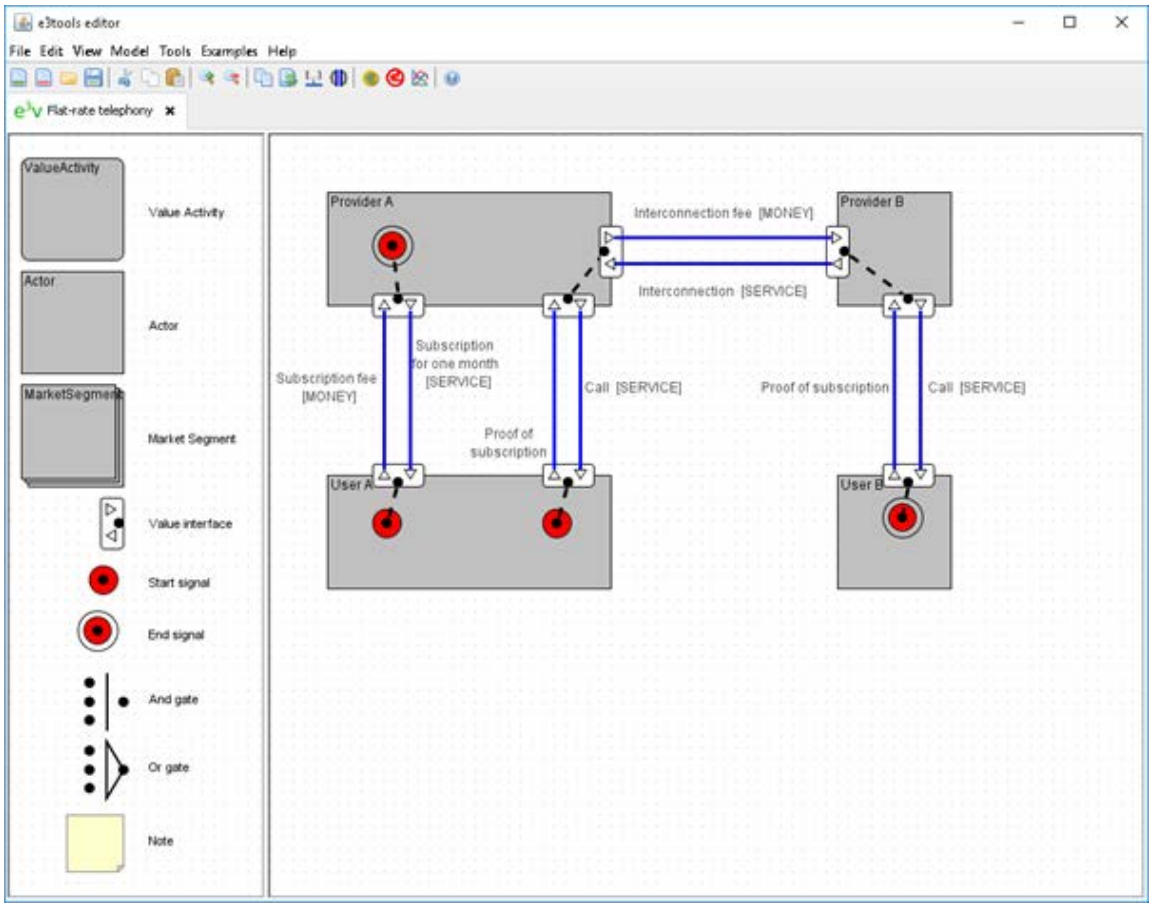

Figure 4.22.: Example of an e3tool value model.

4.10.5.2. Profitability chart

In order to generate a profitability chart, use the entry menu "Tools -> Profitability Chart". This open a dialog asking for actor's perspective to consider for generation. Choose an actor and click on "OK".

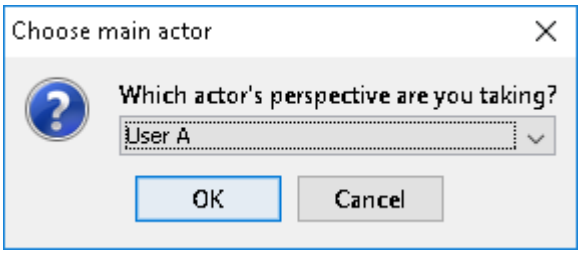

Figure 4.23.: e3tool: choose main actor.

It asks for parameter to use for generation. Choose a parameter and click on "OK".

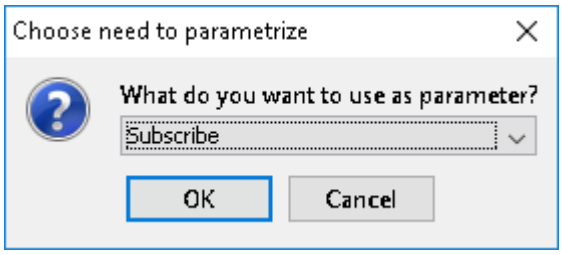

Figure 4.24.: e3tool: choose a parameter.

This finally asks for occurrence rate interval. Enter values and click on "OK".

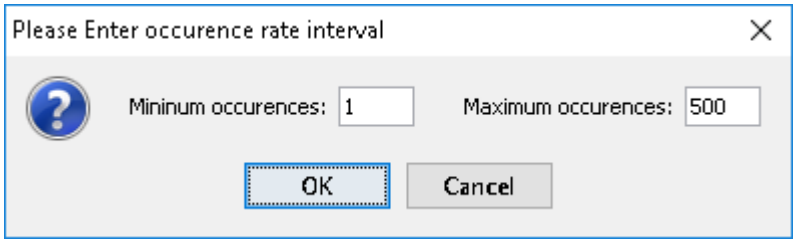

Figure 4.25.: e3tool: enter occurrences.

This opens the profitability chart:

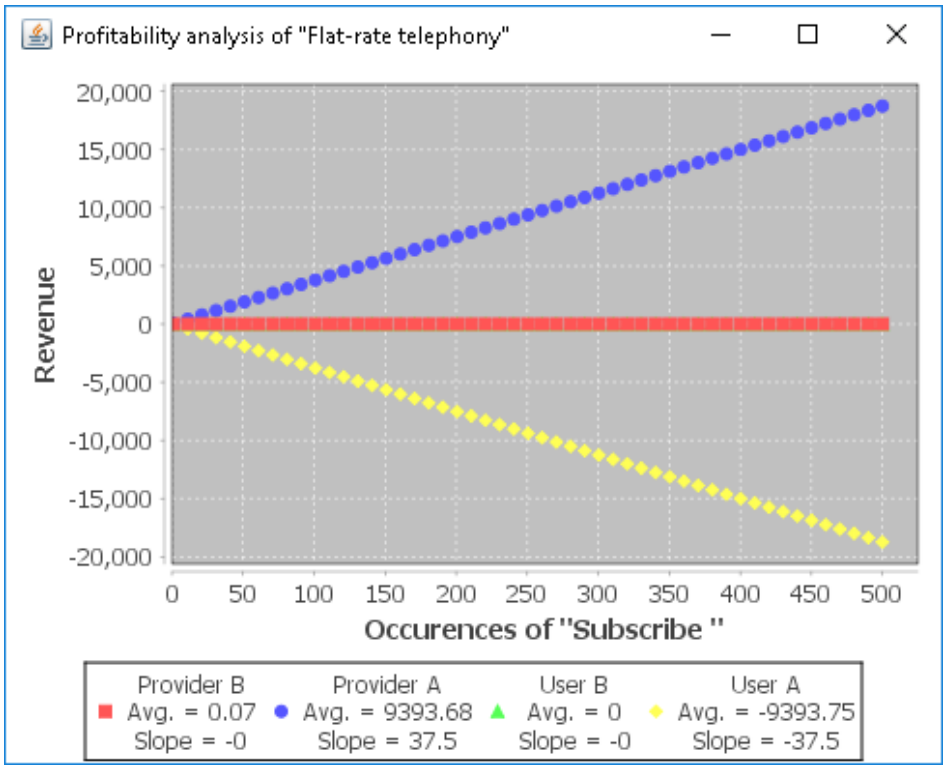

Figure 4.26.: e3tool: profitability chart.

4.10.5.3. Fraud generation

There is an online wiki available at https://github.com/danionita/e3tools/wiki. The wiki contains three tutorials at the moment of writing this deliverable. The wiki also has a page describing in detail how fraud generation is performed.

The tool has usage instructions in the README.md file visible on the project's main GitHub page https://github.com/danionita/e3fraud.

4.11. Attack-Defence Tree optimiser (ADTop)

4.11.1. Description

The Attack-Defence Tree optimiser (ADTop) 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).

Where to find the tool?
https://trespass.itrust.lu/tools in the "Analysis" category.

Current version of the tool: 1.0.

4.11.2. Connection

When you click on the "Run" application link in the TRESPASS platform, a file named "ADTop.jnlp" is downloaded (accept the security message). Double-click on it (accept the execution warning message) to launch the ADTop tool.

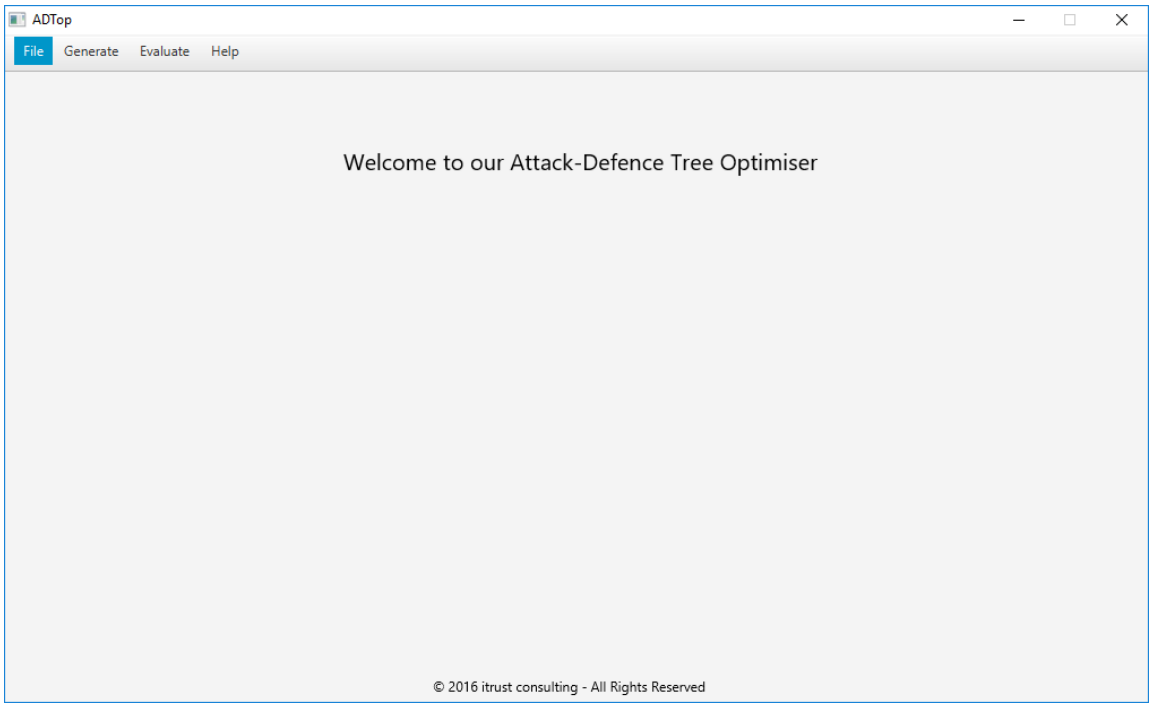

Figure 4.27.: ADTop.

4.11.3. Usage

You can download the three sample input files provided on the TRE$_S$PASS platform in order to test the tool:

ATree.xml.

extract.json.

Association matrix.xls.

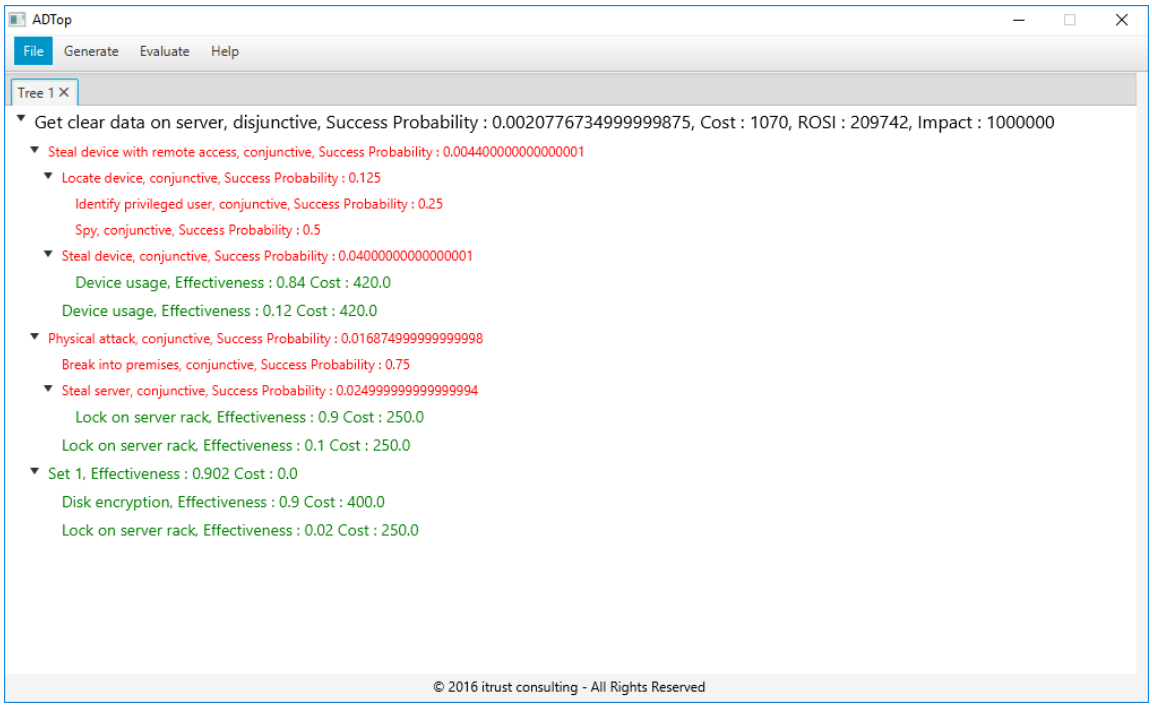

Figure 4.28.: ADTop: Optimal attack-defence tree provided.

See the ADTop user guide attached here for more information:

ADTop user guide.

Additional information is also available in the "Help" menu of ADTop tool itself, by clicking on the "About ADTop..." and "Get started" submenus.

5. Validation of toolchains

5.1. What worked

Chain A APL + ATtop Success Data: APL.xml —> ATtop Time: 2016-10-18 10:35:47 — 2016-10-18 10:35:48

5.2. What didn't

Chain B APL + ATtop + Software Checker Error Data: APL.xml —> Attack Tree Analyzer.xml Time: 2016-10-18 10:44:57 — 2016-10-18 10:44:57

Software Checker in command line was designed at the beginning to work with the IPTV scenario, in order to scan smart TVs using the NMAP protocol. However, this development has not been implemented. Therefore, a tool chain containing Sotware Checker will fail, and it is better for now to use Sotware Checker only as an individual application to check vulnerabilities.

6. Conclusions

During integration and the time of writing this document, several tools and applications did not execute successfully. However, owners of tools and developers altogether did a great job to fix the remaining bugs.

We can say now that all tests made with installed tools were a success.

The TRE$_S$PASS platform was successfully deployed. No bug on the platform itself has been detected nor reported.

The toolchains sometimes failed due to formatting issues or parameters passed between two different tools that need to exchange their input and output files, but these issues are now solved.

This deliverable should be read together with D5.4.2 and D6.4.3.

A. Integration diagram

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. However, the integration diagram gives an overview to the reader on how the applications and tools can interact with each other in order to build tool chains, and where they are located in the map.

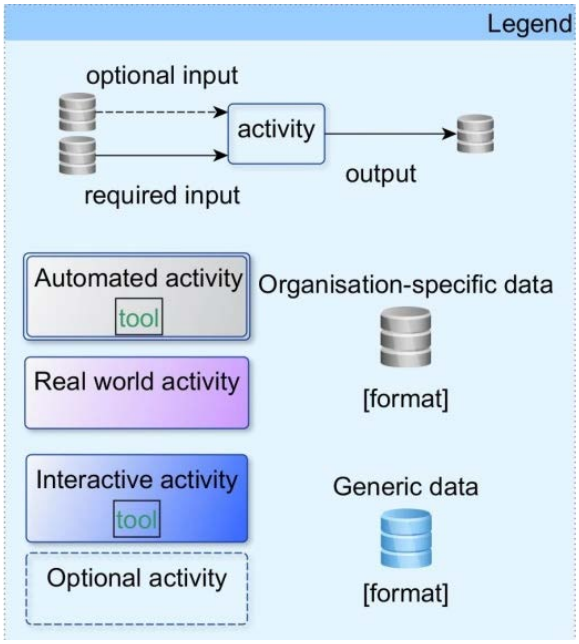

Figure A.1.: Legend for the Integration diagram.

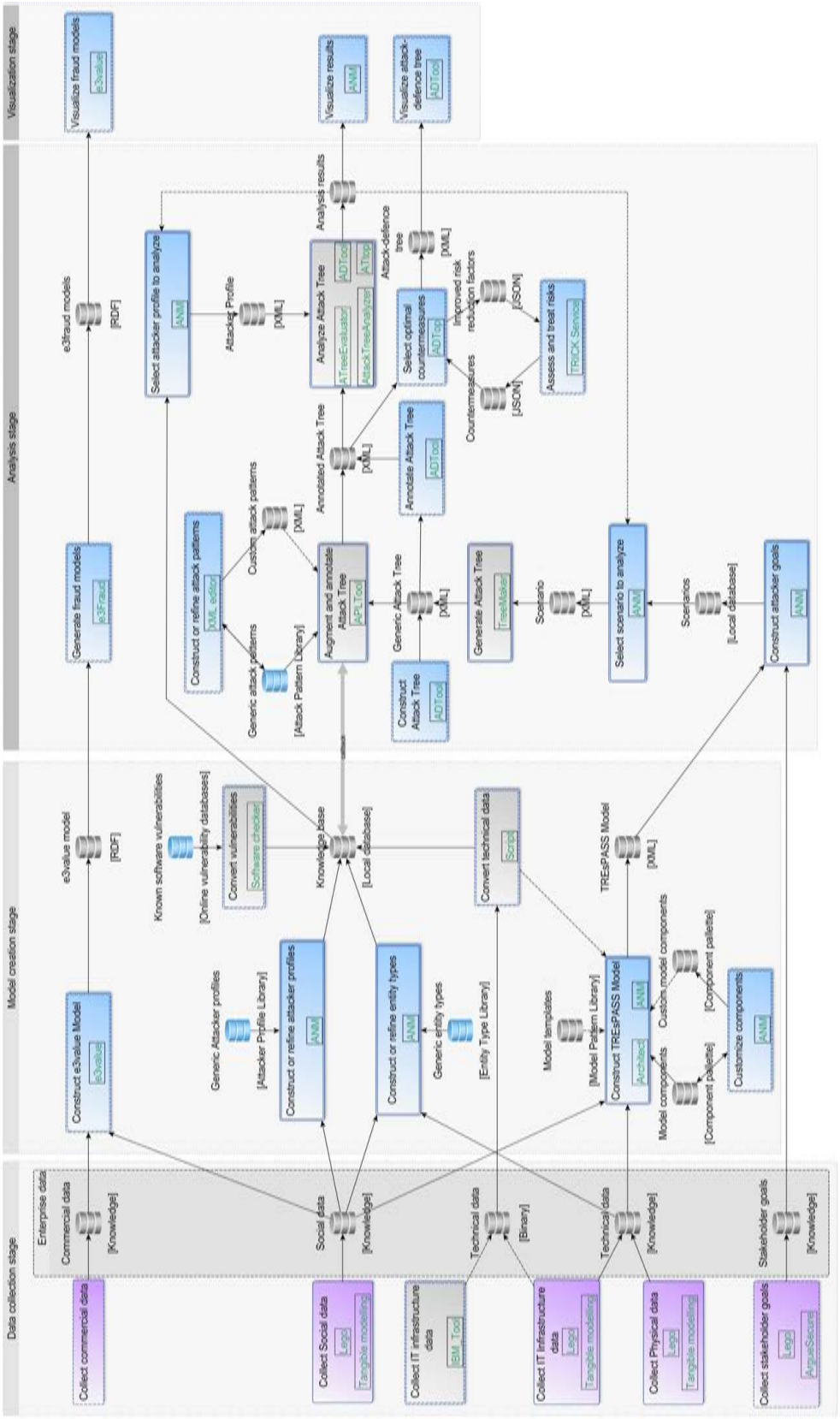

Figure A.2.: Integration diagram for the TRESPASS project.

References

Bistarelli, S., Fioravanti, F., & Peretti, P. (2006). Defense Trees for Economic Evaluation of Security Investments. In *Ares* (p. 416-423). IEEE Computer Society. doi: http://doi.ieeecomputersociety.org/10.1109/ARES.2006.46

Edge, K. S., Dalton II, G. C., Raines, R. A., & Mills, R. F. (2006). Using Attack and Protection Trees to Analyze Threats and Defenses to Homeland Security. In *Milcom* (p. 1-7). IEEE. doi: 10.1109/MILCOM.2006.302512

Ionita, D., Bullee, J. H., & Wieringa, R. J. (2014, August). Argumentation-based security requirements elicitation: The next round. In *Proceedings of the 2014 ieee 1st international workshop on evolving security and privacy requirements engineering (espre), karlskrona, sweden* (pp. 7–12). Piscataway, New Jersey: IEEE Computer Society. http://eprints.eemcs.utwente.nl/25041/.

Ionita, D., Kegel, R., Baltuta, A., & Wieringa, R. J. (2016, September). Argumentation-based security requirements elicitation: The next round. In *Proceedings of the 2016 ieee 1st international workshop on evolving security and privacy requirements engineering (espre), beijing, china.* Piscataway, New Jersey: IEEE Computer Society. (To be published)

Kordy, B., Kordy, P., Mauw, S., Radomirović, S., Schweitzer, P., & Weber, J.-P. (2009–2012). *The ATREES Project, funded by the Fonds National de la Recherche, Luxembourg under grants C08/IS/26 and PHD-09-167.* http://satoss.uni.lu/projects/atrees/. (Accessed December 6, 2013)

Kordy, B., Kordy, P., Mauw, S., & Schweitzer, P. (2013a). ADTool: Security Analysis with Attack–Defense Trees. In K. R. Joshi, M. Siegle, M. Stoelinga, & P. R. D'Argenio (Eds.), *Qest* (Vol. 8054, p. 173-176). Springer.

Kordy, B., Kordy, P., Mauw, S., & Schweitzer, P. (2013b). ADTool: Security Analysis with Attack–Defense Trees (Extended Version). *CoRR*, *abs/1305.6829*.

Kordy, B., Mauw, S., Radomirović, S., & Schweitzer, P. (2012). Attack–Defense Trees. *Journal of Logic and Computation*. (Available at http://logcom.oxfordjournals.org/content/early/2012/06/21/logcom.exs029) doi: 10.1093/logcom/exs029

Mauw, S., & Oostdijk, M. (2006). Foundations of Attack Trees. In D. Won & S. Kim (Eds.), *Icisc* (Vol. 3935, pp. 186–198). Springer. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.97.1056

Salter, C., Saydjari, O. S., Schneier, B., & Wallner, J. (1998). Toward a secure system engineering methodolgy. In *Proceedings of the 1998 new security paradigms workshop* (p. 2-10).

The TRE$_S$PASS Project, D5.4.2. (2016). *The integrated TRE$_S$PASS process.* (Deliverable D5.4.2)

The TRE$_S$PASS Project, D6.4.3. (2016). *TRE$_S$PASS deployment and maintenance plan.* (Deliverable D6.4.3)

The TRESPASS Project, D8.3.2. (2016). *Final IPR management database table.* (Deliverable D8.3.2)