predict
prioritise
prevent

TRESPASS

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

Deliverable D4.2.2

Methods for visualisation of information security risks

Project:	TRE$_S$PASS
Project Number:	ICT-318003
Deliverable:	D4.2.2
Title:	Methods for visualisation of information security risks
Version:	1.0
Confidentiality:	Public
Editor:	Jeroen Barendse
Cont. Authors:	J. Barendse, F. Brodbeck, A. Tanner, P. Hall, L. Coles-Kemp, C. Heath, R. Jhawar, R. Trujillo-Rasua
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
LUST	Jeroen Barendse	1,2,3,4,5,6
LUST	Frederic Brodbeck	3,4
RHUL	Lizzie Coles-Kemp, Peter Hall	1,2,3,4,5,6
RHUL	Claude Heath	6
UL	Ravi Jhawar, Rolando Trujillo-Rasua	E

Quality assurance		
Role	Name	Date
Editor	Jeroen Barendse	2016-09-30
Reviewer	Marianne Junger	2016-10-24
Reviewer	Michael Osborne	2016-10-24
Task leader	Jeroen Barendse	2016-10-26
WP leader	Lizzie Coles-Kemp	2016-10-27
Coordinator	Pieter Hartel	2016-10-30

Circulation	
Recipient	Date of submission
Project Partners	2016-10-01
European Commission	2016-10-31

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

Contents

List of Figures

Management Summary

This deliverable is described in the Description of Work as follows: "Methods for visualisation of information security risks: This deliverable is a refinement of D4.2.1 based on experiences with the prototype in visualising the case studies. It documents the TRE$_S$PASS visualisation approach to visualising information security risks in socio-technical security models, and visualisations to illustrate the dimensions of the risk identified in the data gathering work package." This deliverable reflects the work conducted Task 4.2 and one of two approaches that we have taken in Task 4.3 (with the second approach being documented in (The TRE$_S$PASS Project, D4.3.3, 2016)).

In Task 4.2, WP4 is required to develop "an approach and cartographic visualisations to make explicit security risks in the modelled organisation. The goal is to make risks and their level easily accessible for practitioners." In addition, in conjunction with (The TRE$_S$PASS Project, D4.3.3, 2016), this deliverable also addresses Task 4.3 which is defined in the Description of Work as: "One strand aims at visualising complex technical information in an easy to understand way. The goal is to provide sufficient information to allow users to find, assess, and mitigate risks while maintaining ease of use and simplicity. The second strand will track the development of the socio-technical security model and develop tools to articulate the organisational and social dimensions that affect the likelihood of a successful attack. It will also develop attack/defence trees as a tool to visualise the logic of the model's calculations. These tools are expected to make the model and its calculation much easier to understand for practitioners."

In chapters 2 and 3, this deliverable articulates our visualisation approach by describing:

- The visualisation principles and techniques that we have developed for our work in TRE$_S$PASS;
- The visualisation atlas which provides the visualisation elements used in all the visualisations presented in this deliverable and used within the ANM; and
- The participative visualisation development process that we have developed through our collaboration with practitioner panels and use of paper prototyping.

This deliverable presents our work to visualise the dimensions of risk based on the outputs of WP3. The results of this collaboration with WP3 can be seen in Chapter 4 of this deliverable and in particular in section 4.4 where we present the results of our work to visualise the outputs of the attack tree calculations developed in WP3.

Our strategy for developing usable visualisations can be seen in Chapter 3 and the results of our evaluation programme can be found in Chapter 2. Our strategy for developing usable visualisations was aligned with the iterative development process for the ANM and

underlying tools. In order to keep pace with the overall iterative development cycle, we used rapid prototyping and evaluation techniques found in industry that focus on low-fidelity (particularly paper prototype) testing and focus group led evaluations to provide rapid feedback. We developed a practitioner panel in both the UK and the Netherlands to provide focus group-led feedback during the lifetime of the project. This approach is described in more detail in Chapter 2. In conjunction with the usability evaluation strategy, we also deployed standard HCI techniques for simplifying the presentation of complex data and interaction techniques to encourage user engagement with visualisations; all of which are designed to improve the usability and accessibility of the visualisations. These techniques are described in Chapter 3 and presented in the context of each visualisation tool in Chapter 4.

Key takeaways: In this deliverable we present the main visualisation research contributions to the TRESPASS project, an overview of our main evaluation activities and a summary of our main impact achievements and exploitation opportunities. We have also summarised our achievements in a set of three short books. The title of the set is "Picturing Risk" and these books can be downloaded from https://visualisation.trespass-project.eu/?cat=69

Our project outputs comprise the following elements of the visualisation platform:

- Visualisation toolkit which includes a visualisation atlas and provides a language that can be used to explore new risk areas. We have applied the visualisation language to the Attack Navigator Map (ANM).
- Attack tree visualisation techniques and standalone tools such as the Attack Cloud that tackle the challenges related to attack tree complexity and the communication of risk values in the attack tree form.
- Visualisation techniques to respond to the challenges posed by infrastructural complexity, e.g., in the cloud case where there is a large number of interrelated elements. This is addressed from an end-user perspective in this deliverable and addressed from a model perspective in (The TRESPASS Project, D4.3.1, 2014).

Working in partnership with the information security practitioner community has been central to our visualisation approach. From the end of year one to the end of year four we have undertaken evaluations of the visualisation research outputs presented in this deliverable. In total the outputs presented here have been evaluated by 75 security practitioners and 40 postgraduate students who specialise in applied security studies. These evaluations took place from the end of the first year of the project to the final month of the project. As part of the evaluation process we ran four security practitioner panels and eight security practitioner paper prototyping sessions. We then mirrored the paper prototyping sessions with three postgraduate student evaluation sessions. Finally we concluded our evaluation process with an evaluation by eight security practitioners.

In addition to our evaluation programme, we have conducted four large engagement sessions during this project where we have both demonstrated and transferred the risk visualisation knowledge we have gained during this project. In year three of this project we applied the visualisation principles and techniques developed in WP4 to the Verizon Data Breach report and, as we demonstrate, achieved impact within the lifetime of this project. Furthermore, we have an impact and exploitation programme underway that will see the outputs of the TRESPASS visualisation work live on beyond this project. We document our engagement, exploitation and impact efforts at the end of this report.

1. Introduction

This deliverable, together with (The TRE$_S$PASS Project, D4.3.2, 2016), and (The TRE$_S$PASS Project, D4.3.3, 2016), present the main outputs of the visualisation work package. All three deliverables present responses to the need for understanding and assessing threats which has always been a critical challenge to security practitioners. This challenge is exacerbated by the rise of the digital era and the increasingly intricate systems that make up the information security landscape. As a result, developing methods that distill vast amounts of data into consumable visualisations or diagrams that are accessible, engaging and informative remains a critical issue. Many security models[1] can be thought of as giant machines with dozens, even hundreds of levers and dials that must all be precisely calibrated in order to model each specific security scenario. The machine metaphor captures the behind-the-scenes complexity of socio-technical models for security and risk estimation, but obscures the goal of achieving usability: the "general rule" of visualisation being that the user stays in control and that the computer "offers choices with appropriate feedback for user actions" (Bederson, 2003).

Visualisation is used, not merely for aesthetics, but also to aid practitioners and end users in forming mental models by providing a visual aid for the data presented in a security model. The data presented within models such as Attack Trees (Schneier, 1999), however, tends to be complex and tedious to analyse. It is therefore the focus of the visualisation effort within TRE$_S$PASS to:

- Develop a visual language for communicating the results of the analysis performed using the WP3 tools to target user communities, as defined by the wider project.
- Develop techniques to respond to a number of the complexity problems that emerge in this context.

1.1. Goals

In line with the focus outlined above, the primary goal of WP4 is to develop a set of processes for identifying, developing and validating visualisations of information security risks.

[1] A model is defined as “a simplified description, especially a mathematical one, of a system or process, to assist calculations and predictions” (Oxford English Dictionary). When discussing visualisation of said models, it is in regards to making this abstraction visible in some manner.

The core visualisation requirements (The TRESPASS Project, D4.1.2, 2015) have been interpreted in the context of a general TRESPASS visualisation platform. The term "visualisation platform" covers a conceptual space that enables risk visualisations to be created and adapted. The platform also provides a space for the sharing and creation of visualisations, visual languages, visual abstractions and methods of evaluation. These visual methods and tools are brought together in the TRESPASS Attack Navigator, which has a project-wide user interface termed the TRESPASS Attack Navigator Map where all tools developed within the project can be viewed, accessed, and connected.

1.2. How to access the prototypes

Prototypes are at the core of the three final deliverables from WP4. There are several ways to access these prototypes.

1. Via the visualisation prototypes, tools and methods showcase:
   https://visualisation.trespass-project.eu (no log-in needed).
   This showcase includes:
   - Publications
     - Book series '"Picturing Risk":
       https://visualisation.trespass-project.eu/?cat=69
   - Methods for visualising security risks
     - Attack Tree component visualiser (XML to Attack Tree):
       https://visualisation.trespass-project.eu/?p=409
     - Visualisation Atlas:
       https://visualisation.trespass-project.eu/?p=122
     - Attack Cloud visualisation:
       https://visualisation.trespass-project.eu/?p=236
     - Attack Tree visualisations:
       https://visualisation.trespass-project.eu/?cat=5
     - TRESPASS security visualisation methods:
       https://visualisation.trespass-project.eu/?cat=12
     - ATM case study visualisation:
       https://visualisation.trespass-project.eu/?p=117
     - DBIR Attack Graphs 2015 visualisation:
       https://visualisation.trespass-project.eu/?p=275
   - Complexity prototypes

  - Time-Containment Visualiser (TiCoVis):
    https://visualisation.trespass-project.eu/?p=55 (no log-in) or
    https://trespass.itrust.lu/tkb/tkb/TiCoVis (log-in at the TRESPASS portal required - see below)
  - Complexity prototype Cloud Environment Actor Visualiser (CEAV):
    https://visualisation.trespass-project.eu/?p=216 (no log-in) or
    https://trespass.itrust.lu/tkb/tkb/CEAV (log-in at the TRESPASS portal required - see below)

- Social-technical visualisations
  - InterActor:
    https://visualisation.trespass-project.eu/?p=482
    The above page on the showcase includes a link to the prototype itself [2]:
    http://104.131.113.255:3000/
  - Paper Prototyping:
    https://visualisation.trespass-project.eu/?cat=14
  - Lego methods:
    https://visualisation.trespass-project.eu/?cat=18

2. Via the TRESPASS portal:
   https://trespass.itrust.lu/login.
   First time log-in: Click on Sign-up, you will receive a confirmation email, you need to click on it to acknowledge the registration. The itrust ICT administrator will have to personally validate your account. Once you receive the validation email, you will be able to access with the same credentials:
   - The individual tools
   - The Attack Navigator
   - The Attack Navigator Map
   - The SVN repository for the update of programs
   - Visualisation components
     (directly accessible at https://trespass.itrust.lu/visualisations)

Getting started with the prototypes

- Downloadable demo file: ATM case study XML (can for instance be used in the Attack Tree component visualiser (XML to Attack Tree):
  https://visualisation.trespass-project.eu/?p=409

[2] Please note that isolated issues have been reported accessing the prototype using *Eduroam* wifi in certain institutions, due to local access rules at host institution sites.

- Github repositories where visualisation code is hosted:
  https://github.com/trespass-project
- The manual to the Attack Navigator Map:
  ANM Manual

1.3. Document structure

Chapter 2 introduces the visualisation platform that has been developed in WP4, summarises the critique of attack tree visualisations that led to the development of the visualisation platform and discusses an example of where greater flexibility in attack tree visualisation is needed. Chapter 3 discusses visualisation principles and techniques. Chapter 4 shows TRE$_S$PASS visualisation innovations and how the principles and techniques from Chapter 3 are applied in TRE$_S$PASS. Chapter 5 is concerns WP4 engagement, impact and exploitation, and in that chapter we present the main engagement activities conducted by WP4 and summarise the short-term impacts and the potential for longer term impacts and exploitation. A summary of our most significant contributions to the state of the art in risk visualisations can be found in our concluding chapter, Chapter 6.

1.4. Foreground and background

The following elements of the visualisation platform are foreground IP:

- the Attack Navigator Map;
- the visualisation toolkit;
- a visualisation process;
- a TRE$_S$PASS visual language;
- the attack tree visualisations;
- trespass.js
- the visualisation techniques to respond to the challenges posed by infrastructural complexity;
- the strategy for evaluating TRE$_S$PASS visualisations.
- the interface to the Attack Navigator.

Other libraries and frameworks, that are distributed under open source or creative commons licenses are background to their respective developers and include D3[3], Angular[4], Bootstrap[5], jQuery[6], lodash[7], react[8], and Docco[9].

[3] see Data-Driven Documents http://d3js.org.
[4] see AngularJS http://angularjs.org/
[5] see Bootstrap http://getbootstrap.com/
[6] see jQuery http://jquery.com/
[7] see Lo-Dash http://lodash.com/
[8] see react https://facebook.github.io/react/
[9] see Docco http://jashkenas.github.io/docco/

2. Visualisation in TRESPASS

This chapter introduces the visualisation platform that has been developed in WP4, summarises the critique of attack tree visualisations that led to the development of the visualisation platform and discusses an example (complex systems) of where greater flexibility in attack tree visualisation is needed.

2.1. Visualisation platform

The term *visualisation platform* covers a conceptual space that enables risk visualisations to be created and adapted. The platform is a space where visualisation tools can be shared and contributed. It also provides a space for the sharing and creation of visualisations, visual languages, visual abstractions and methods of evaluation. These visual methods and tools are brought together in the TRESPASS Attack Navigator which has a project-wide user interface where all tools developed within the project can be viewed, accessed, and connected. This visual user interface also provides access to a set of exploratory tools developed as part of the visual platform to explore new risk areas.

The focus in this deliverable is on the general visualisation principles and techniques developed in TRESPASS to visually articulate the risk landscape and the visualisation of the attack tree form (the form chosen in WP3 for the analysis of information security risks). In so doing, this deliverable responds to both Task 4.2 and Task 4.3 and addresses aspects of visualising socio-technical risk visualisation and complexity. In particular, as part of its contribution to visualising the social aspect of information security risk, this deliverable demonstrates how the social dimensions of attack tree analysis results have been visualised and the approaches to visualising attacker profiles. Particular focus to the visualisation of social networks and a framework for adding social data to the TRESPASS model is presented in the deliverable (The TRESPASS Project, D4.3.3, 2016). Whereas the focus on visualisations to extend the state of the art in the visualisation of complex risk scenarios for the cloud environment is presented in (The TRESPASS Project, D4.3.2, 2016).

The visualisation platform has been created in response to the requirements identified in (The TRESPASS Project, D4.1.2, 2015):

- A visualisation must have a particular goal: simplify the process of developing security-related visualisations, and develop convincing visualisations.
- A visualisation must be usable and accessible: develop a toolkit to simplify the user generation of security visualisations.

- Develop a language, toolkit and processes for the articulation of different types of security visualisations.

The visualisation platform provides the visualisation tool kit and contains the following components:

- Visualisation tools: a family of approaches that includes digital and analogue tools, including attack trees, automated data stream visualisation tools, LEGO , and other interrelated components of the TRESPASS visual language.
- Methods of visualisation evaluation.
- A visualisation process.
- A TREsPASS visual language described as part of collection of visual concepts and tools termed the Atlas).
- Attack Navigator Map, which represents a target system in cartographic form, displaying the different types of connection between the elements of the system.
  - The Attack Navigator Map builder, where users can construct, import and build an attack model.
  - The Attack Navigator Map visualisation dashboard, where attack models created in the Attack Navigator Map builder are analysed by various tools (AT Evaluator, AT Analyzer, ATtop) and visualised as attack scenarios.
  - Attack tree visualisation tools, including: AD tool and Converter.

Many early version of tools have been described in earlier deliverables: (addressed in (The TRESPASS Project, D4.1.1, 2013) and followed in (The TRESPASS Project, D4.2.1, 2014) and (The TRESPASS Project, D4.3.1, 2014)). This deliverable reviews their current state as concept and working prototypes.

2.2. Survey of current approaches to attack visualisation

One of the first steps of the TREsPASS project was a survey of state-of-the-art information security risk visualisations (The TRESPASS Project, D4.1.1, 2013). In general, information security visualisations depend very much on purpose (exploratory versus explanatory), topic (financial risks, environmental risks, computer security etc) and target audience, with very different levels of abstraction in presenting the vast amounts of data typically available for the systems under consideration. Therefore they range from dashboard-like presentations of the overall system state for awareness in an operations centre, to tools for investigation of very specific technical details, such as packet flows across networks, for deep diving into available data.

This breadth makes it difficult to survey the complete field. Some summarising reviews can be found in (Roth, 2012), (Eppler, Martin J. and Aeschimann, Markus, 2008), (Husdal,

2001) for more general risk visualisations and (Marty, 2008) for the more technically-oriented visualisations in computer security. Since (The TRESPASS Project, D4.1.1, 2013) discussion of the challenges of visualising complexity and uncertainty has proliferated across disciplines from the use of visual analytics to assess uncertainty and value impact in aeronautical engineering (see (Kipouros, 2016), to the use of a honeycomb structure visualisation system in security for "situational awareness" of large-scale networks (Park, 2014). A significant development in the epistemology of visualisation is in Drucker (Drucker, 2014) which explicates the theory that visual forms of knowledge production are distinctive in their own right, rather than being perceived as a secondary, representational tool of data. Drucker's argument that "what is seen is what is made" suggests that a visualisation of an information network in fact constructs and shapes our understanding of its behaviour, rather than simply reflecting what is there. Finally, (Roth, 2012) lays the groundwork for the participatory approach to visualisation that has been developed in TRESPASS, arguing that risk communication can benefit from visualisation tools that allow the user to explore their own perceptions of risk and vulnerability, and also "contribute data to the visualisation – making it a more dynamical two- way process" (Roth, 2012). Such approaches have become more frequent in recent community-based projects in participatory mapping of flood and bush-fire risks (Akama & Ivanka, 2010)

Here we focus on a critical review of tools currently used by security practitioners such as Carnegie Mellon's OCTAVE (Alberts & Dorofee, 2002) and Siemens' CRAMM (Barber & Davey, 1992). These findings, which serve as background and motivation for a new approach, are summarised here.

Information security visualisations have traditionally been used to display degree of impact, measure of risk, and value of assets. Tools similar to those previously mentioned, use visualisations that map assets to threats and vulnerabilities and appear often in dashboards. These visualisations cover a wide range of graphic outputs, including visual metaphors to convey certain portions of their security model. However, in most cases, visualisation approaches focus more significantly on functional implementation and interaction rather than the narrative defined by the visual choices. Our work seeks to address that gap.

2.3. Our work to extend the state of the art

The state of the art as described above presents tools that typically depict the information security attack surface as having only two parameters, requiring researchers to sometimes oversimplify a model in order to represent it. However in doing so, crucial interrelationships between actors and elements of the system are omitted and there is a risk of misrepresenting the data or portraying it in a way that causes the viewer to misinterpret it. To counter this risk, we have explored methods of extending existing visualisations to support higher dimensionality, and to allow viewers to switch between different perspectives of the same data, so that previously hidden connections and entities can be brought to the surface. A visualisation language was developed that supports representations of individual details of the model as well as the model in its entirety.

An example of this approach can be seen in the visualisations that we created for the Verizon DBIR report. In this activity we used the visualisation language to develop arc diagrams of multivariate data on breaches, drawing from the work of (Wattenberg, 2002). Arc graphs represent connections by placing the nodes on the same line, allowing for easy comparison of nodes and edges. The two sided nature of the DBIR arc graph allows us to differentiate between attributes (on one side) and actions (on the other). We also used semantic zooming to demonstrate how certain attacks paths might be structured, as well as the relative frequency of certain action or attribute categories within the attack space.

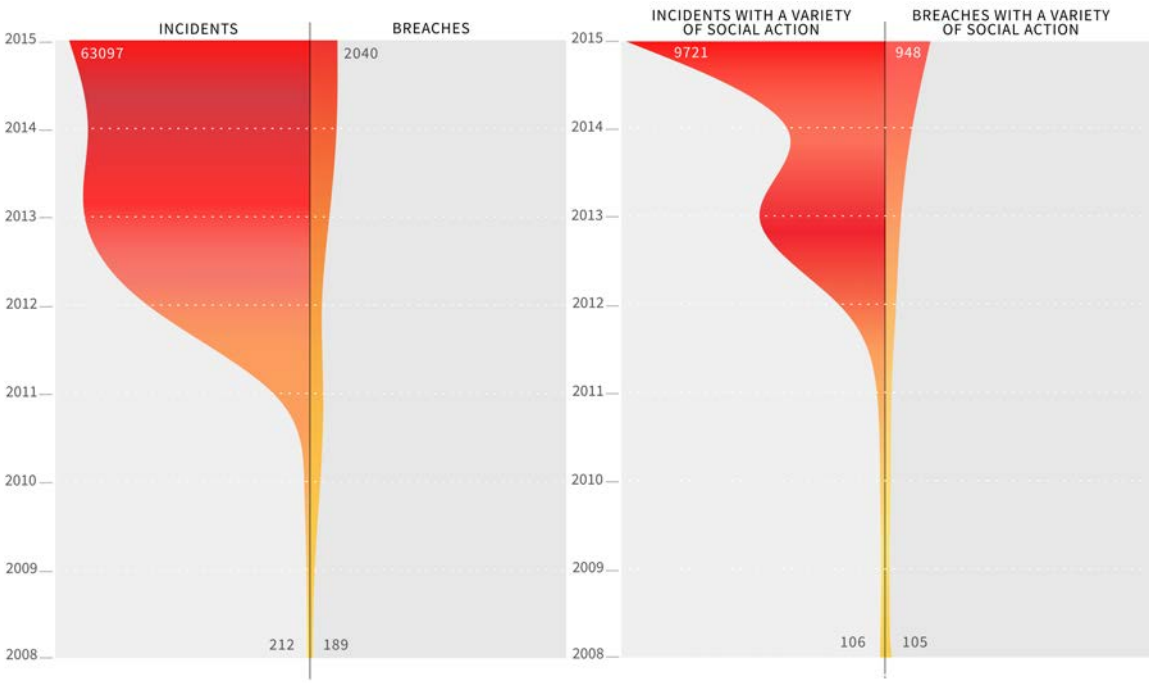


Figure 2.1.: Left: Visualisation of incidents and breaches from 2008—2015.
Right: Visualisation of incidents and breaches with a variety of social action from 2008—2015 as gathered for the DBIR Report. Social actions can be fishing, baiting, bribery, scam, extortion, propaganda, spam, and so on. Data courtesy of the DBIR Report, Verizon, 2015.

WP4 has also extended the state of art in terms of the ability of risk visualisations to respond to complexity. Complexity refers to the number of possibilities in the problem space (Flach, 2012). As discussed in (The TRE$_S$PASS Project, D4.3.2, 2016), the characteristics of complex systems are high-dimensionality, non-linearity, sensitivity to initial conditions, adaptivity and emergent behaviour. Stappers and Norman (Norman & Stappers, 2015) develop the discussion of complexity within a design framework with the term Design-X, which describes complex societal systems such as healthcare and government policy implementation where there are social and technical components 'whose interactions are critical to the system's overall behaviour.' Organising the properties of Design-X problems into three categories, psychological, social and technical, Stappers and Norman provide

an account that describes well the visualisation challenges of socio-technical security. The nine properties of complex, socio-technical Design-X problems are:

The psychology of human behaviour and cognition

- System design that does not take into account human psychology.
- Human cognition: the human tendency to want simple answers, decomposable systems, and straightforward linear causality.

The social, political, and economic framework of complex socio-technical systems

- Multiple disciplines and perspectives
- Mutually incompatible constraints

The technical issues that contribute to the complexity of Design-X problems

- Non-independence of elements
- Non-linear causal relations: feedback
- Long and unpredictable latencies
- Multiple scale sizes
- Dynamically changing operating characteristics

2.4. Evaluation

Evaluation is an important aspect of the TRE$_S$PASS visualisation programme. The usefulness of the visualisations to the security practitioners in the field needs to be understood in order to ensure that the visualisation platform is adequate and meets the needs of the TRE$_S$PASS stakeholder communities. In total we have evaluated our visualisation research outputs with 75 security practitioners and 40 postgraduate students specialising in applied security studies.

An agile, rapid prototype testing process was chosen for evaluation where we tested collections of visual elements with small groups of users to gain rapid insights and feedback into the prototype development cycle. This approach suited the TRE$_S$PASS project because of the programme of on-going updates to the Attack Navigator Map (ANM) and the evolutionary nature of the development of the prototype analysis tools. The approach to usability testing also followed this approach and used techniques commonly found in industry for testing the usability of early prototypes: namely paper prototyping and focus group evaluation. In particular, the visualisation atlas was evaluated using paper prototyping methods by the practitioner panels and by specific evaluation feedback sessions. In-line with industry standards, participants were provided with situated tasks and asked to evaluate the usefulness, expressivity and usability of the visualisation elements.

Initially, agile process for visualisation development was chosen as the most appropriate given the method, given the iterative nature of the TRESPASS development process in general. However, TRESPASS developed, we realised that the visualisations were representing aspects of a complex adaptive system and visualisations of this complexity have no single, universal meaning. It therefore became a deliberate part of the visualisation philosophy to evaluate the visualisations situated within contexts of use to better understand the usability, expressivity and usefulness of each development.

Overview of the rapid prototype testing
Sometimes referred to as 'low-fidelity' testing (Boling & Frick, 1997), this is the standard method for evaluation when working in a fluid and dynamic development environment, coupled with think-aloud activities (Jaspers, Steen, van Den Bos, & Geenen, 2004) where participants talk about the interface experience as they work with either the digital or paper prototype.

At the core of the rapid prototype process were evaluation sessions with groups of 5-8 security practitioners, repeated over several sessions with different groups of practitioners, in line with the standard size of between 8 and 20 participants in a feedback sample (Faulkner, 2003). These sessions were termed 'practitioner panels'. An evaluation session took place on average every four months and feedback was provided on both the visualisations and the underlying TRESPASS concepts. These sessions took place in the Netherlands, Australia and the UK.

The evaluation process used paper prototyping, the presentation of wire frames, hands-on interaction with digital prototypes and discussion and feedback sessions.

In each session the same process was followed, and the steps were as follows:

1. Introduce the TRESPASS project and the role of visualisation within the project.
2. Present participants with a scenario and, where appropriate, a paper prototype kit with which to map the scenario and explain how to use this analogue mapping kit.
3. Ask participants to identify the assets, the connections between the assets and the possible attack paths.
4. Place a likelihood on the success of each attack (represented by an attack path).
5. Rank the risks based on the likelihoods.

The results were recorded through photography, note-taking and the collection of the completed paper prototyping (Fig. 2.3).

The paper prototyping method Paper prototyping is a means of creating a paper version of a digital interface and inviting a participant group to engage with the paper prototype, simulating the way in which they would interact with a digital interface. This method can be used to gain insights into user attitudes and expectations with regard to any such interface, which can then be translated into designs for digital prototypes that take account of these expectations and usages.

Four main paper prototyping sessions have been undertaken: two paper prototype sessions took place in Australia, one session in Brussels and one in the UK. A mapping kit was developed for these sessions (Fig. 2.2). This is composed of:

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

2.5. Insights gained

Since the early stages of the TRE$_S$PASS project, the aim in WP4 has been for paper and other physical modelling approaches to be linked with digital prototypes, so that they may inform each other in a reciprocal fashion. This leads in turn to new iterations of prototype designs.

The key insights resulting from the evaluation sessions with paper prototyping have been as follows:

- Narratives support greater understanding of the map.
- Clear categories of risks, visually identified, help to make the map usable.

This led us to consider including narrative in the Attack Navigator Map. Innovation in this area requires the incorporation of analogue three-dimensional modelling (such as *LEGO*), and other forms of data concerning risk perceptions, into the more mathematically abstracted Attack Navigator Map, as discussed above.

Each practitioner panel had at its core the evaluation of:

- The colour palette used in each visualisation
- Form and deployment of icons and symbols
- Stacking of objects and shapes within the visualisations presented

Each practitioner panel also addressed the visualisations as collections of visual elements used to communicate a particular narrative and participants were asked to comment on both the visualisation as a whole as well as the individual elements. Below are summaries of the main practitioner evaluation panels that we have run during this project.

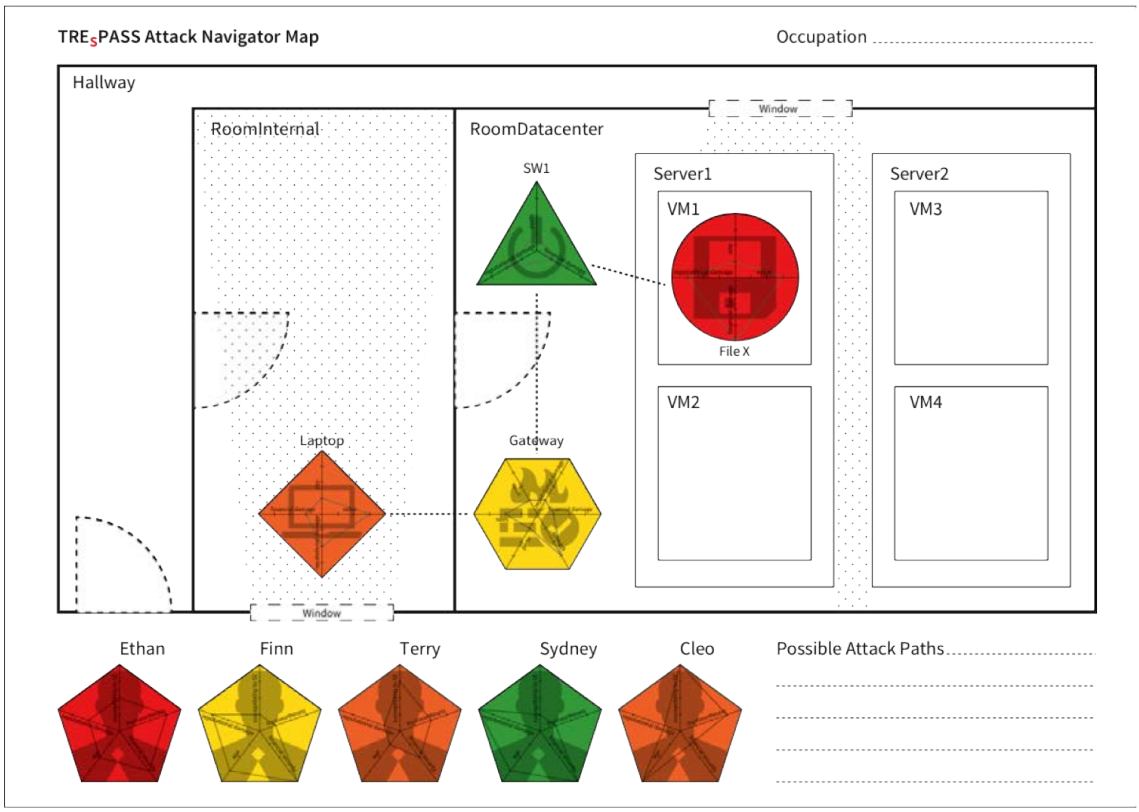


Figure 2.2.: The graphical template that was used for paper prototyping.

2.5.1. Dutch practitioner panels

In 2015 three Dutch practitioner panels evaluated the visual elements. These panels took place in March, June and November 2015. The practitioners represented security consultants and practitioners in security consultancies and large commercial organisations in the Netherlands.

The same evaluation format was used in each panel:

- Present wireframes and digital prototypes of the current visualisations.
- Present vignettes of risk analysis using the current visualisations.
- Provide opportunities using paper-prototyping techniques for participants to explore the risk visualisations.
- Discussion and feedback on the visualisations provided.

The feedback from the second and third panel was in-line with the feedback from the first panel and reflected the fact that participation of between 8 and 20 participants in total provide the majority of the insights to be gained from larger forms of assessment (Faulkner, 2003).

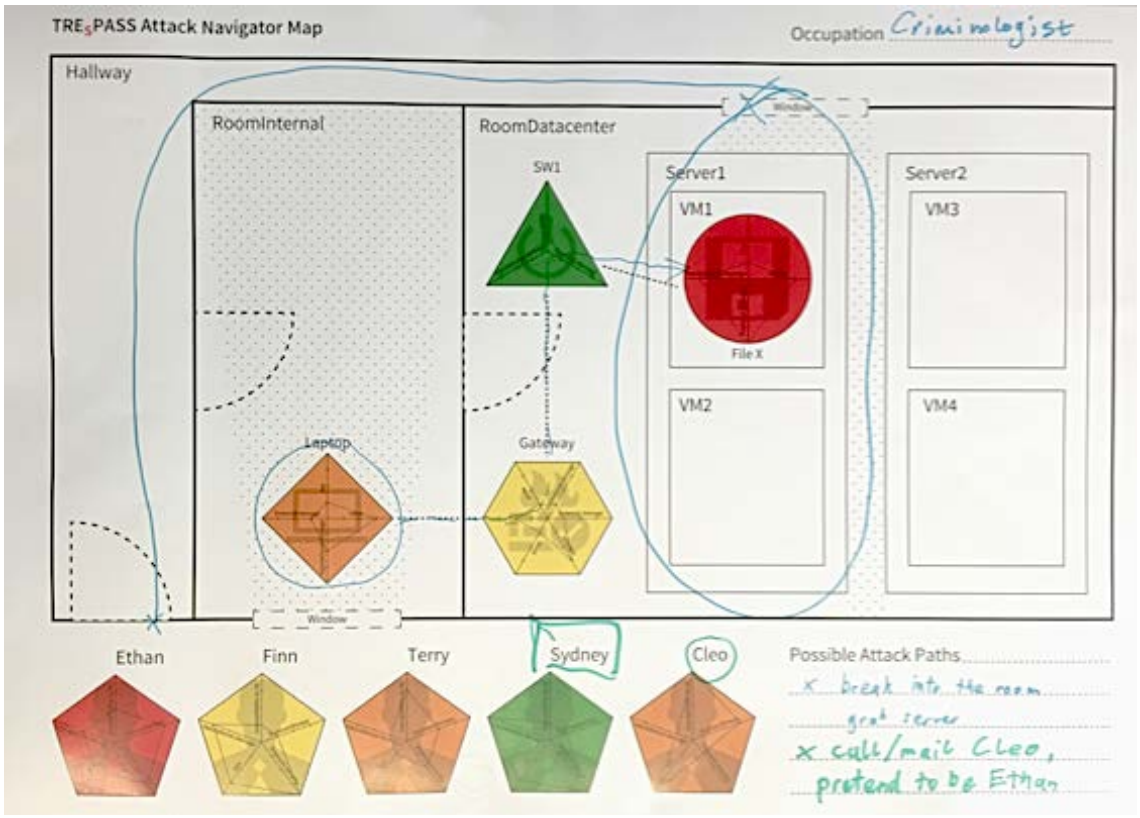


Figure 2.3.: The completed template, a sample result from the paper prototyping session at CSP, Brussels.

The feedback from the panels shaped key ways in which the visualisation principles and techniques were applied in TRESPASS's case. A number of key insights were put forward in these panels that shaped the TRESPASS visualisations.

- It was less useful to visually place the social, physical and virtual space in one map. Consequently separate maps were produced and subsequent evaluations confirmed that the separation of maps were more beneficial to the analyst and made the ANM visualisations more usable.
- The original colour scheme was muted and the participants felt that this was very "70s" in look and feel. Consequently we developed a wider colour palette and deliberately introduced colour as a means of drawing the eye to the results of the risk analysis.
- We used spider diagrams to represent relationships between variables but the panels' input led us to desist from using this form and to concentrate on the use of colour and line thickness instead, as this approach was regarded by the practitioners to provide better expressivity.

The feedback gave us a steer as to which narrative should be brought out in different visualisation scenarios. These insights helped us to develop different views of the ANM

visualisations which have been implemented to aide usability and accessibility of the visualisations. For example, one participant stated:

> "Showing all possible attack paths is not very useful. From a business point of view the top ten would be enough. Visualisation helps most if I can for instance export it and take to the next meeting with my supervisor / manager."

Another participant commented in the context of the Cloud scenario:

> "For instance for the Cloud case study, physical location of data is either unknown or irrelevant. In such case, access control is much more interesting, maybe focus more on those aspects."

All panelists emphasised that risk impact should be the focus of visualisations as this is crucial to an analyst's activities. As a result of this type of insight, where impact is addressed by the WP3 tools, the visualisations ensure that it is highlighted using colour and saturation.

The panelists highlighted that security risk assessment starts with the asset and then considers asset protection. As a result, the ANM uses maps that are asset-centric so that practitioners have a degree of familiarity with the visual layout.The importance of asset protection was also fed back to the wider project team and the insights contributed to the use of attack-defense trees in the ATM case study.

2.5.2. Australian practitioner panel

An Australian practitioner panel was run in March 2015 as part of the Bridge Point Forum. Four groups of six security practitioners took part in a rapid paper prototyping session to evaluate the use of colour, symbols and element stacking in TRE$_S$PASS visualisations.

The practitioners were presented with a risk scenario and were asked to build a risk map from that scenario using the paper prototyping toolkit that WP4 had developed. (For details on the paper prototyping toolkit please refer to (The TRE$_S$PASS Project, D4.1.2, 2015)).

The feedback from this session was as follows:

- Risk scenarios are essential for the effective construction of a risk map and therefore the visualisation process should target a particular risk scenario.
- The paper prototyping kit included the use of spider diagrams in order to map multiple risk variables to an asset. The spider diagrams proved difficult to use and the participants did not find it a helpful way to display risk variables. This tallied with the feedback from the Dutch practitioner panels.
- The interpretation of the risk map was seen to be heavily influenced by the mix of stakeholder backgrounds in the group. This point is reflected in the UK practitioner panel's insight that creating stakeholder specific views help the map's legibility. In subsequent practitioner panels we therefore aimed to gather further inputs as to which views would be most beneficial.

2.5.3. UK practitioner panels

A UK practitioner panel was run in June and October 2016.

In June five information security practitioners were consulted on the visualisations used in the Attack Navigator Map and the TRE$_S$PASS physical modelling processes.

The focus of this panel was to look at the visualisation techniques of the use of symbols and element stacking as deployed in the ANM.

Three out of the five participants interpreted the TRE$_S$PASS visualisation capability as a variety of means for recording risk discussion sessions and turning the outputs of these sessions into collective models. The focus for these participants was on the legibility of those collective models. These participants described the TRE$_S$PASS attack navigator maps as a type of mindmap that can both calculate and articulate the potential range of attacks in a given context. In this narrative the practitioners recognised the flexibility of the TRE$_S$PASS visual language that extends the state of the art in mindmaps by allowing more explicit and typed modelling, including arbitrary types for nodes and edges, as well as having provision for automated executions to generate attack trees and perform risk calculations from this model.

This panel focused on the use of symbols and element stacking as a means to improving legibility of the maps. The following feedback was given:

- The use of views was endorsed as a means to provide the information that is relevant for a particular stakeholder.
- The symbols were intelligible to the participants and the maps were correctly interpreted. It was felt that no further action was needed to adjust or add to the iconography used in the ANM.
- It was felt that further clustering and element stacking could be used to reduce the number of visual elements on a map in order to make the map easier to interpret.

2.5.4. Rapid paper prototyping sessions

In April 2015, we ran eight parallel paper-prototyping sessions with security practitioners at the CSP event in Brussels. In this session, the paper prototyping was used as a mean of creating a paper version of a digital interface and inviting a participant group to engage with the paper prototype simulating the use of the digital interface. While working on the paper prototypes, it was emphasised that there are very different viewpoints possible onto the scenario, namely organisational, physical, digital and social.

The session followed the process to introduce the TRE$_S$PASS project first, present a concrete scenario and the mapping kit. Participants were then asked to identify important assets in the scenario, their relations and possible attack paths. Attack paths should be evaluated with their perceived likelihood of success to enable a ranking of the different attacks.

The results, as recorded through photography, note-taking and the collection of the completed paper prototyping, showed that: (a) Narratives (e.g. risk scenarios) are needed to make maps understandable; (b) Risks need to be visually categorised in order to make the map usable; and (c) There was a tendency among participants to focus on left-side parts of the map compared to parts on the right-hand side, suggesting a general bias, at least in the West, to read from the left.

These results tally with the feedback from the Brisbane security practitioner panel.

More details (including pictures) can be found in (The TRESPASS Project, D4.1.2, 2015).

2.5.5. Postgraduate evaluations

Three postgraduate paper prototyping sessions were run in the UK (1 session) and Perth, Western Australia (2 sessions).

In the Australian evaluations, students were highly engaged both in the task – particularly in understanding the parameters of the activity and construction of the use case, and in group discussion negotiating what risks were present, the strength of these risks, and how to calculate the magnitude of any risks present. Further, the activity promoted the interpersonal communication and team work required to build consensus within the group. Photographs and note taking followed by discussions with students after the sessions constituted the data collection process.

Analysis of the photographs of the group results, review of the notes taken during the session together with discussion with the groups during the activity revealed the following general observations:

- The mapping of the strengths and weaknesses using the spider diagram was not undertaken by most of the groups (tallying with the feedback from the Dutch and Australian practitioner panels).
- The modelling of the assets and actors was well understood and the symbols were clearly intelligible to each group.
- The concept of using height of the colouring to reflect the magnitude of the vulnerability was not embraced or explored by any group. This led us to examine how colour and size was used in this respect and to revise the HCI principles used.
- The delineation between the physical and digital space was well understood by the end of the activity, although some students had difficulty conceptualising and representing the difference between the physical and digital assets. For instance, where a server holds the data to be protected was not easily classifiable as either physical or digital. This response reflected the value of adopting the input from the Dutch practitioner panels on the topic of separating the maps but also demonstrated to WP4 the importance of clearly articulating the relationships between the different types of spaces.

The UK postgraduate session focused on the relationship between the risk scenario and the attack navigator maps produced. For the first time we trialled the combined use of the physical modelling and the attack map form to see how the two forms might best work together. The students conducted a physical modelling exercise whereby a risk vignette was created in LEGO by each group and then the group produced an attack navigator map using the paper prototype kit to reflect the attack paths within the vignette.

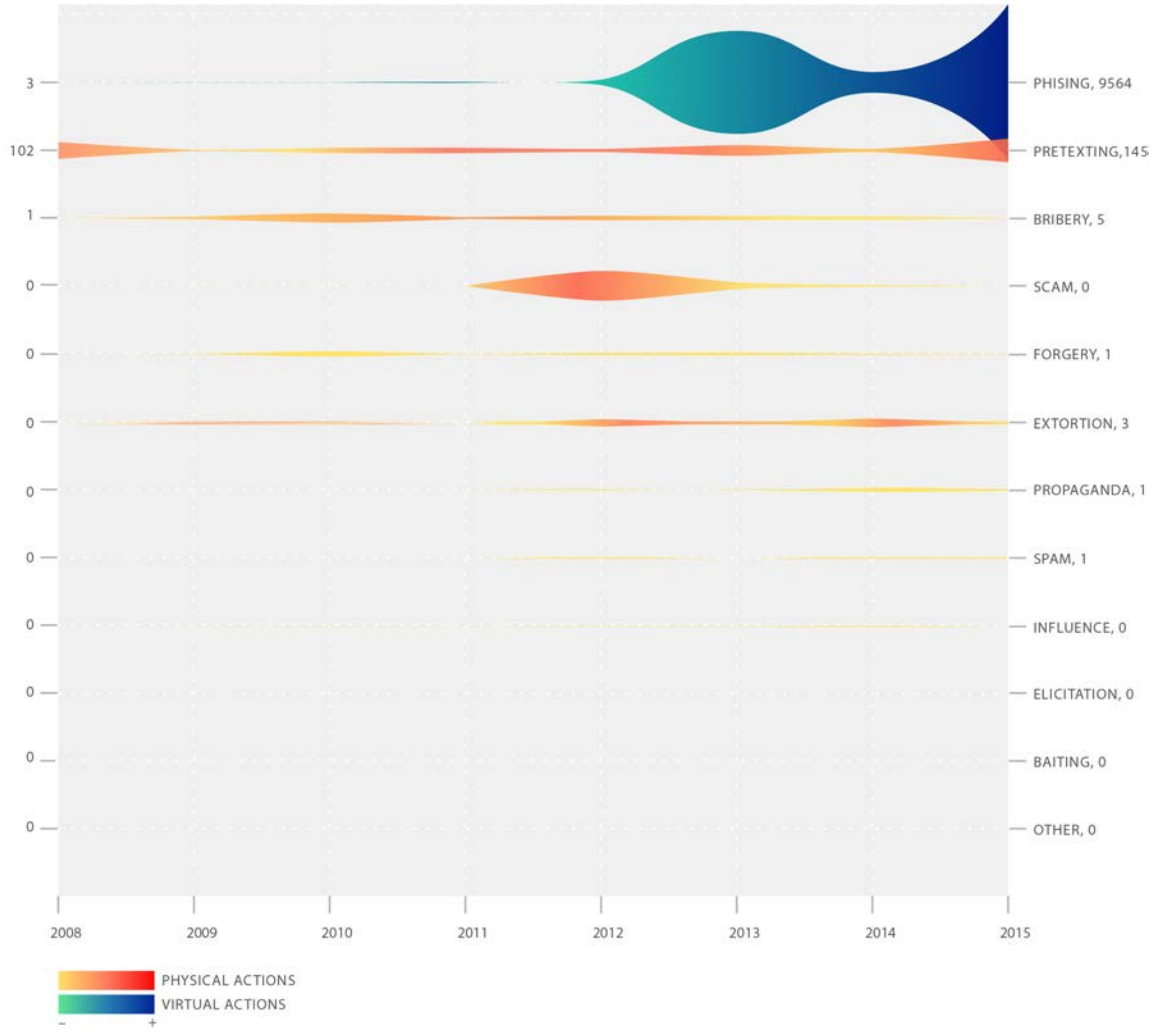


Figure 2.4.: Visualisation for social action variety by year for incidents. Any similarity of the visualisation of the data on "fishing" to resemble a fish is coincidence, but does help a viewer to remember the data. Data courtesy of Verizon.

3. Visualisation principles and techniques

WP4 has based its visualisation approach on a number of generally accepted visualisation principles that are used within the current state of the art in interaction design, many of which contributing to the usability of the visualisations. For WP4, these are grouped under the headings of gestalt, iconography and interaction principles. The three groups are described in the following sections.

3.1. Gestalt and visual thinking

The core of any visualisation is the selection and development of an effective visual vocabulary and a mapping, or legend, that supports it. Such visual vocabularies are often aided by the principles of *Gestalt* psychology.

The overall appearance and qualities, or *Gestalt*, of a visualisation are important properties. *Gestalt* is a term from psychology defined as the 'unified whole'. Being aware of and implementing the principles of *Gestalt* theory in a visual language can have the effect of making visualisations feel stronger and more coherent. These theories of visual perception were first developed by a group of German psychologists (Koffka, 1935)(Koffka, 1922) in the 1920s and describe how people tend to organise visual elements into groups. Although there are certain faults with some Gestaltist assumptions (Ware, 2000), it is important to be aware of those principles in order to use and, at other times, also to creatively mis-use them:

Similarity The principle of similarity states that things sharing visual characteristics such as shape, size, colour, texture, value or orientation will be seen as belonging together.

Continuation The principle of continuity predicts the preference for continuous figures.

Closure The principle of closure applies when viewers tend to see complete figures even if part of the information is missing.

Proximity The principle of proximity or contiguity states that things which are closer together will be seen as belonging together.

Figure and ground The terms figure and ground explain how viewers use elements of the scene which are similar in appearance and shape and group them together as a whole. Similar elements are contrasted with dissimilar elements (ground) to give the impression of a whole.

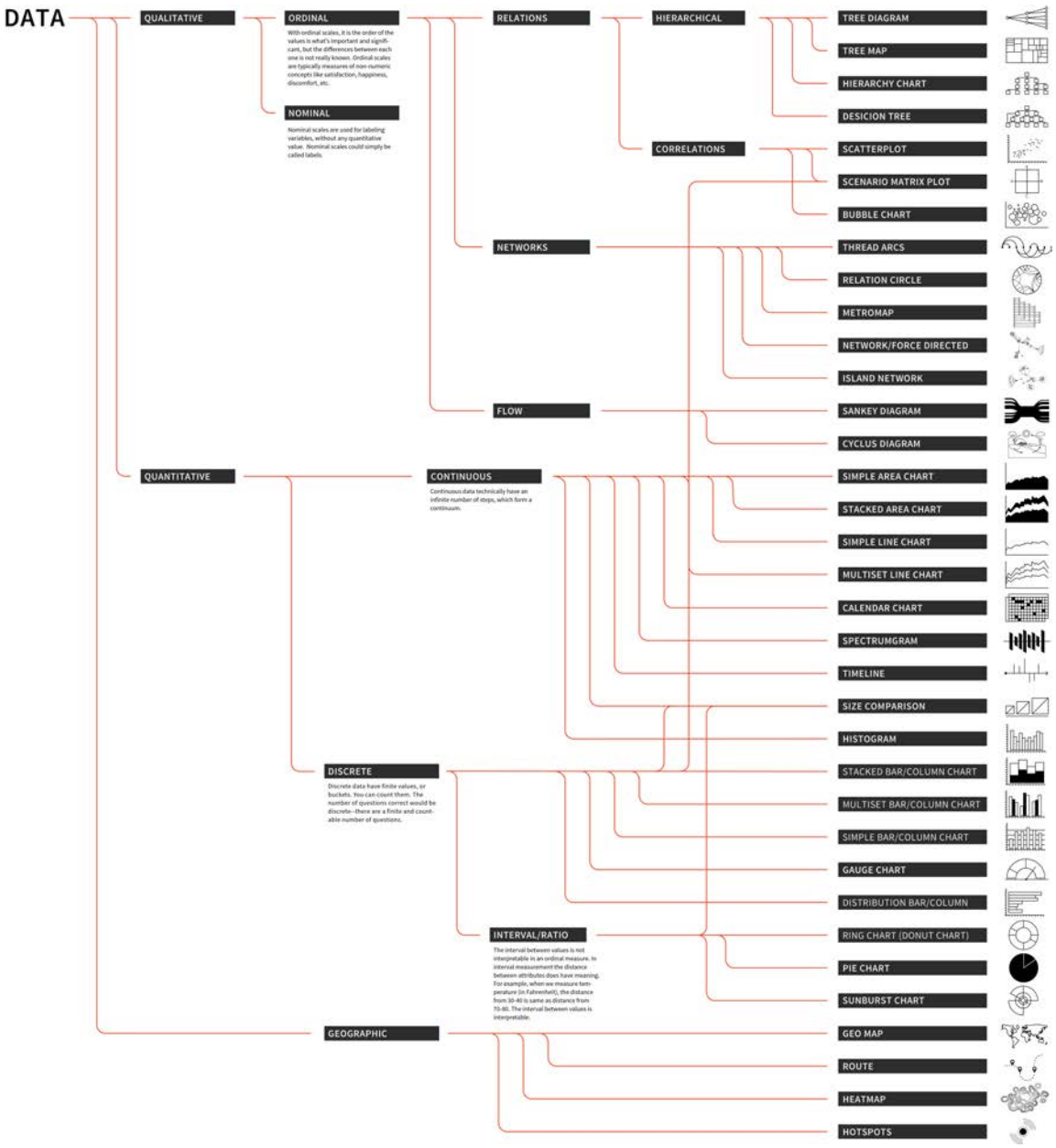


Figure 3.1.: Decision scheme for choosing the correct visualisation for the given data, available as one of the visualisation tools in the Attack Navigator.

Pre-attentive variables and layering Pre-attentive variables operate mostly at a 'sub-conscious' level; people recognise trees, tables, and maps, and immediately process the underlying data according to the first impressions gained without any conscious analysis of actual data. Encoding via pre-attentive factors relates to the general graphic design concept of 'layering'. When looking at well-designed graphics of any sort, different classes of information are perceived on the page. Pre-attentive factors like colour cause visuals to perceptually 'pop out,' and any sense of similarity causes

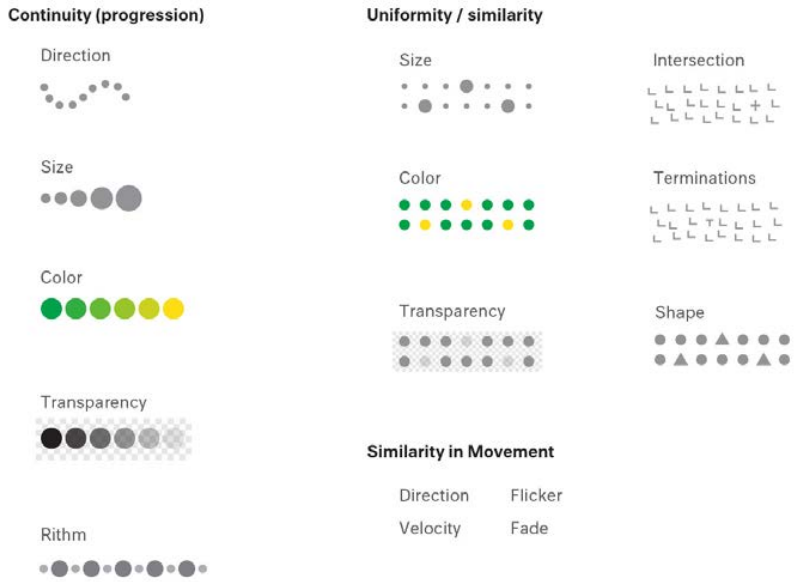


Figure 3.2.: Continuity and Uniformity. Part of the Gestalt section of the Visualisation Atlas, LUST, 2013.

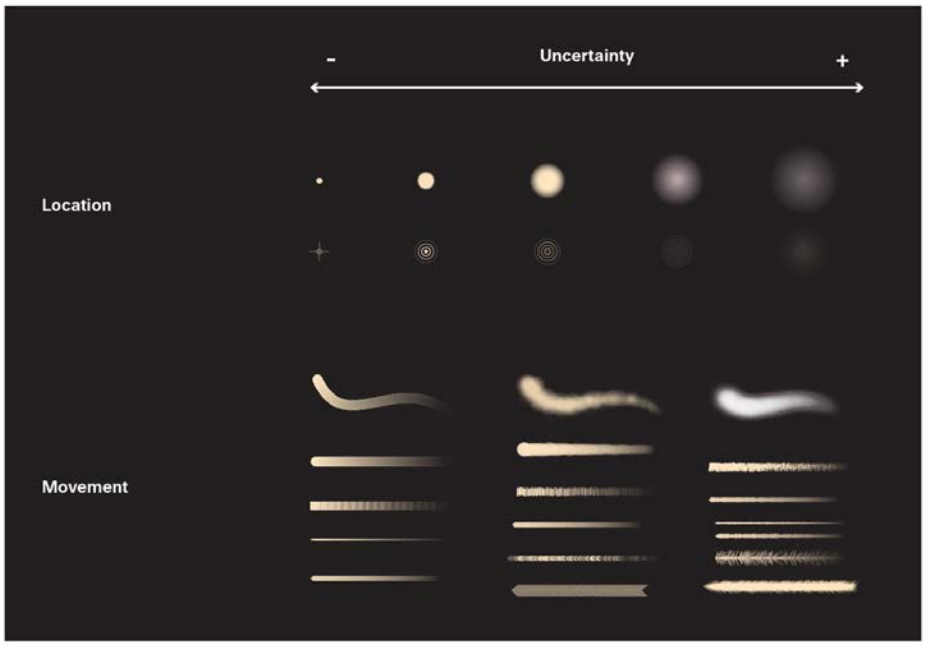


Figure 3.3.: Visualisation of uncertainty. Part of the Gestalt section of the Visualisation Atlas, LUST, 2013.

them to be seen as connected to one another, as if each were on a transparent layer over the base graphic. This is an extremely effective way of segmenting data, where each layer is simpler than the whole graphic, and the viewer can study each layer in turn, while relationships among the whole are preserved, emphasised, and therefore are brought seamlessly to the analyst's attention. There are eight pre-attentive variables: shape, color hue, color brightness, position, orientation, color saturation, texture, and size (Tidwell, 2005).

Pre-attentive variables, combined with certain cultural habits (the colour red indicates stop or dangerous), can already lead to a basic understanding of a visualisation by viewers (Figure 3.4). Therefore it is important that these pre-attentive variables and habits correspond to rather than contradict the mapping chosen. As (Tufte, 1990) said strikingly '. . . avoiding catastrophe becomes the first principle in bringing colour to information: Above all, do no harm.' This certainly applies to visualisation of security related information. Colour used well can enhance and clarify a presentation. Colour used poorly will obscure, muddle and confuse. While there is a strong aesthetic component to colour, using colour well in information display is essentially about function: what information we are trying to convey, and how (or whether) colour can enhance it.

A general rule for visualisations for TRE$_S$PASS is that colours are used to convey a meaning, and the specific colour indicates the extent to which an object or item needs attention.

3.2. Iconography principles

The use of icons (symbols or ideograms that convey their meaning through their pictorial resemblance to a physical object), can be an effective way for users to distinguish various elements. The age-old adage that a picture is worth a thousand words, is often true with icons. TRE$_S$PASS tools and visualisations can benefit from the use of pictograms and iconography as icons can often transcend languages. This is why road signs and similar pictographic materials are often applied as global standards expected to be understood by nearly all. As (Parish, 2007) remarks, visual language is by definition polysemic and by nature more successful in conveying specific objects ("door", "key") than abstract ideas. However, more abstract ideas can be expressed in icons, although those icons are more open to interpretation. Visualising an idea will often support its understanding.

For instance, privacy online is usually predicated on lengthy privacy policies regarding the use of cookies, web beacons, targeted advertising, and when the issuing organisation might share your information with law enforcement. Icons can be also be an effective way to visualise such (little-read) policies, as (Raskin, 2010) shows in an experiment where a 3000 word privacy policy text could be abbreviated to three pictograms.

In developing an iconography for TRE$_S$PASS we have deployed the following principles:

- Where possible, icons are drawn from one visual style.
- Icons are always accompanied by a supporting text label to aid clarity.

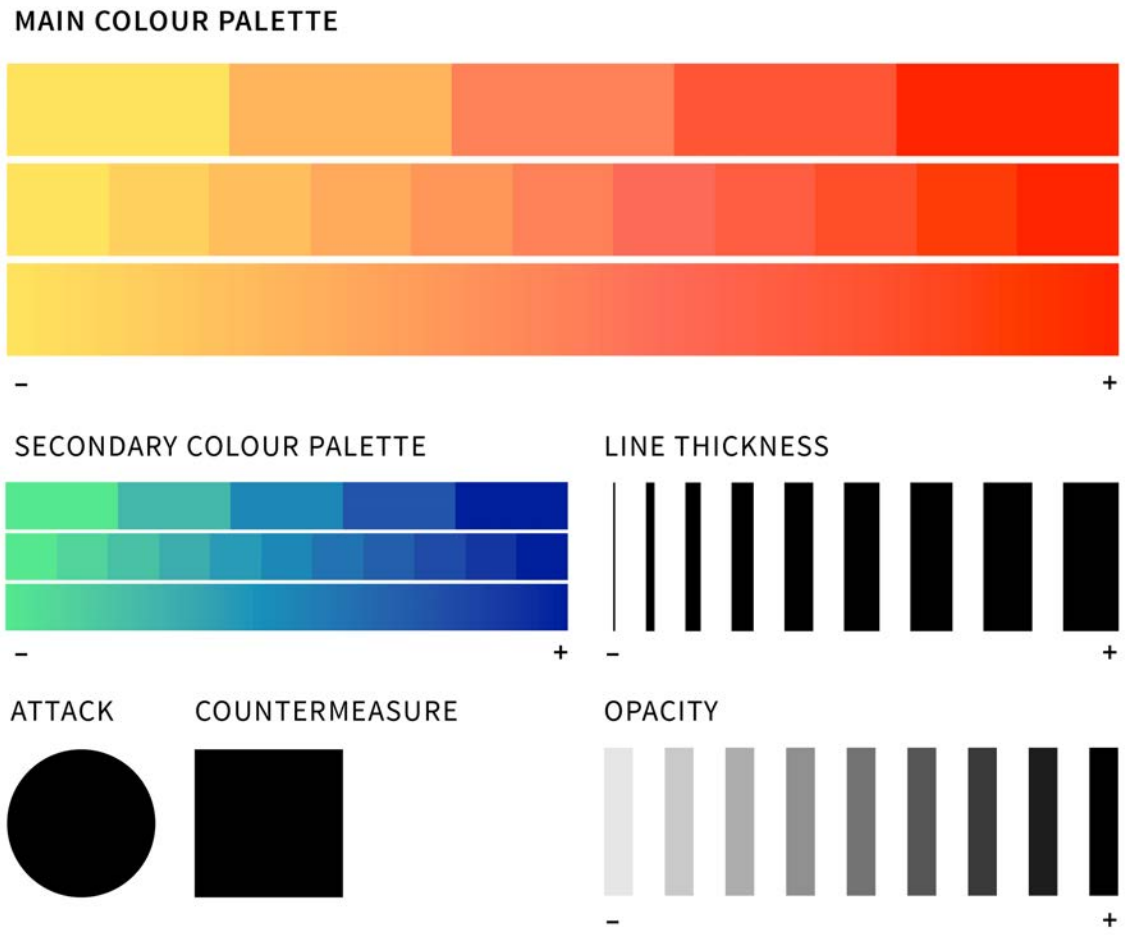


Figure 3.4.: TRESPASS basic building blocks: a main colour palette and a secondary colour palette, which can be used for instance to differentiate countermeasures, or to differentiate physical and virtual steps. In general, the further the colour is to the right of the scheme, the more attention it needs (for instance, deeper red signifies that the item is more vulnerable). The various line thicknesses can be used to express various parameters, as well as the opacity of these lines. Almost all elements from this basic building blocks can be combined, so line thickness with colour, or opacity plus line thickness

- Once the icon set is developed, lesser used icons are replaced by general icons to reduce icon set size.

Developing the icon set

The icons developed were presented in survey form to a small feedback group of 15 people consisting of designers, students, coders, and programmers, with for each icon a minimum of four variations. The results of the survey led to the choice of the final icons, and informed decisions on how to adapt them to better communicate their purpose. The icons are used in the Attack Navigator Map but also in, for example, geo-location-based visualisations of the ATM case study.

Figure 3.5.: A selection of the icons developed for the TRESPASS project. They are used in the Attack Navigator Map, but also for instance in geographic visualisations for the ATM case study.

3.3. Interaction principles

One of the first taxonomies on interaction tasks for information visualisation was presented by (Shneiderman & Plaisant, 2003). It consists of the following seven basic interaction tasks: gaining overview, zooming, filtering, getting details-on-demand, viewing relationships, getting access to a history of actions and extracting sub-collections and query parameters. (Yi, ah Kang, Stasko, & Jacko, 2007) subsequently developed a taxonomy consisting of eight categories of tasks which connects users' goals with basic interaction techniques and profoundly helps the usability of the interactions:

1. *Selecting items of interest:* For graphs, this means marking single nodes and edges, but also sub-graphs or whole diagrams of a model.
2. *Exploring the visualised dataset:* Navigating from one part of the visualised graph to another should be as easy as possible. In existing tools this basic method of graph navigation is usually realised by zoom+pan mechanisms. Another method is to provide techniques for exploring a model for example by quickly moving from one diagram to another.

3. *Reconfiguring the user's perspective:* In terms of graph visualisation, this might mean rearranging the layout of a graph. In this way, hierarchies, clusters or nodes with many links can be perceived more easily. The challenge here is to provide a smooth and understandable transition from one layout to another. This is important, as the layout heavily contributes to the user's mental model.
4. *Encoding the data in a different representation:* This implies transforming the graph to a completely different visualisation such as an adjacency matrix or a list which sorts nodes according to particular metrics. Another way to change the encoding is to use additional attributes such as colours and rendering styles of nodes and edges to emphasise particular properties.
5. *Adjusting the level of abstraction of a data representation (Abstract/ Elaborate):* In the context of graph exploration, this means, for example, showing the properties of nodes and edges at different levels of detail. This can be beneficial to get an overview of large graphs or if users with different goals and expertise are involved. Visualising several levels of abstraction can also be applied to a whole model. For example, diagrams showing content on a coarser level can be linked to diagrams for representing the low-level view.
6. *Filtering the dataset according to specific conditions:* For graph exploration this means filtering particular nodes and edges, for example according to their type, size, semantics etc. (Shneiderman, 1996).
7. *Showing connections between data items:* Visualising connections by nodes linked with each other is an inherent characteristic of graphs (Frisch, 2012).
8. *Query parameters:* An important means for users to understand the nature of the interaction they are experiencing.

Another useful interaction with information can be a *search function*. It enables the user to go straight to the information, regardless of hierarchy. A common downside of a search is the lack of context, although this could be solved partially by how the presented information is designed.

Relationship between interaction and gestalt
We consider interaction an inherent part of the gestalt of a visualisation, especially when dealing with lots of data points and complex systems. Useful approaches to handle the visualisation of such complex systems and support the usability of such visualisations include:

Filtering/highlighting/sorting these functions can be used to select a subset of elements to reduce visual clutter; similarly, sorting of elements allows one to restrict the focus on a subset, utilising a metric for the purposes of ranking

Exploiting visual form and representations utilise visual form and well-known representations to allow quick and high-level recognition.

Using abstractions use abstractions in the set of elements to allow grouping of 'similar' elements and combining them into fewer elements in order to visualise effectively.

Overview and drill-down give an overview of the total system, possibly starting with higher-level abstractions of subsystems, while allowing drill-down into individual subsystems to show more detail.

Multiple views show multiple views of the system from different viewpoints or 'gazes' to highlight different aspects of the system at the same time in a *coordinated-visualisation* (North & Shneiderman, 2000).

3.4. Developing TRESPASS specific principles

'[...] user interaction should not be underestimated in terms of its contribution to perception of information and it should therefore be considered a key part of any visualisation system.' (Kalawsky, 2009)

In (The TRESPASS Project, D4.1.1, 2013) we distilled a data design process of five steps based on Ben Fry's (Fry, 2007) universal process, combined with the narrative-centered design concepts of (Mazza, 2009):

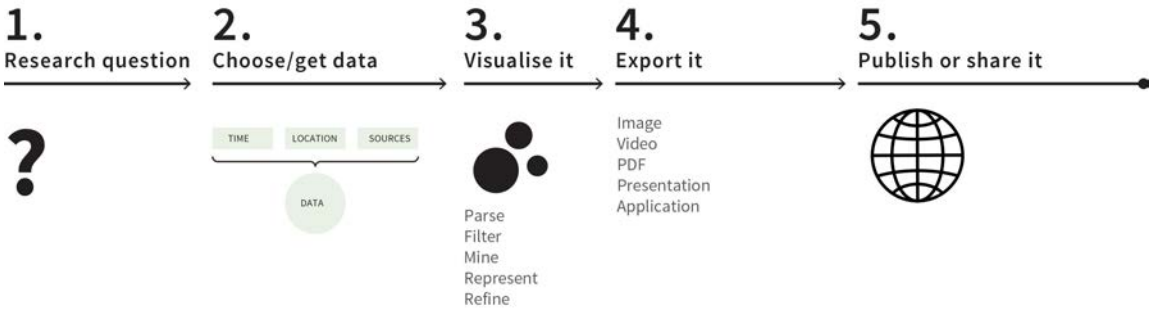


Figure 3.6.: The five step data design process.

In addition, our general visualisation approaches include the principles listed in the previous section. However, we also need to consider the focus of the narrative and the nature of the target user communities. We have therefore further applied Riccardo Mazza (Mazza, 2009), principles:

- *Problem:* What is the main purpose of the visualisation? Is it needed for reporting purposes, used for exploring the dataset in order to find new information, or is the purpose to confirm an assumption or prove a hypothesis?
- *Data type:* What type of data needs to be visualised? Is it nominal, ordinal, interval, or ratio data? (Mazza combines interval and ratio types under the label quantitative.)
- *Number of dimensions:* How many dimensions need to be examined using the visualisation? These are defined as the number of independent attributes (the attributes that vary with respect to one or more independent attributes).

- *Primary structure of data:* What is the structure of the data we need to visualise? Are there simple values, or are we primarily interested in temporal aspects of the data? Is the data of spatial (physical extents), hierarchical, or network structure? Are we interested in a distribution of values?
- *Type of interaction:* How much interaction is needed for the task? Can we use a static display? Does the user need to be able to transform the data prior to visualisation, or manipulate display attributes like colour or zoom-level?

3.5. Guidelines for designing TRESPASS visualisations

The range of data visualisation types spans simple bar charts to real-time, highly interactive data visualisations that display ten data sources and its interpretation. In order to streamline TRESPASS visualisations, we set a few simple base rules that improve every visualisation.

- Keep it simple. Always choose the simplest way to convey your information.
- Have a specific message you want to communicate. Identify the relationships and patterns of your data and focus on what you want to show.
- Maximise the 'data ink' ratio (Tufte, 1990): Data-ink is the non-erasable core of a graphic, the non-redundant ink (or pixels) arranged in response to variation in the numbers represented. In normal words, use "ink" to represent data, not for decoration.
- Select the appropriate chart form (Fig. 3.1) and know its strengths and limits.
- Use colour to highlight vulnerabilities and threats. Use size, and position to help the reader see what is important as size and position draw attention to particular data points and show hierarchy. Colour adds emphasis, highlights particular data points, and draws connections between graphs. Consider that the hue, value, and intensity of the colour are significant and may have cultural or social connotations. Cultural reading conventions also determine how people read charts.
- Use clear and understandable headlines and labels to describe the take-away message of the visualisation.
  - If there is a zero line, add elements that make clear what is below zero, and what is above. Colour and lines are usually good for this.
  - Try to highlight the important numbers on the X and Y axis.
- Pay attention to the legend, which provides viewers with a means of quickly reading and analysing the risk scenario.
  - Designing a legend is also a means to determine which elements should be in a legend and which should not. Less is more in many cases.

  - Labels should be placed as close to the data as possible (see Figure 3.7) to give the reader access to the information.

- Add hierarchy to your data. Avoid cherry-picking data, but do not treat all data equally. Data should have an order of importance. Design choices help communicate this hierarchy.
- Offer multiple views on the data if possible. Different people prefer different representations. Each representation offers unique perspectives on the data.

The following project-specific rules that should be applied to visualisations in TRESPASS:

- Typeface is always Source Sans (mostly regular and semi bold).
- Use uppercase for labels; this supports clarity. If possible, add some spacing between the letters.
- A colour palette that uses a stepped gradient from yellow to red indicates how vulnerable something is. This colour palette can be supported by a secondary palette to indicate other elements like countermeasures or to highlight socially engineered elements.

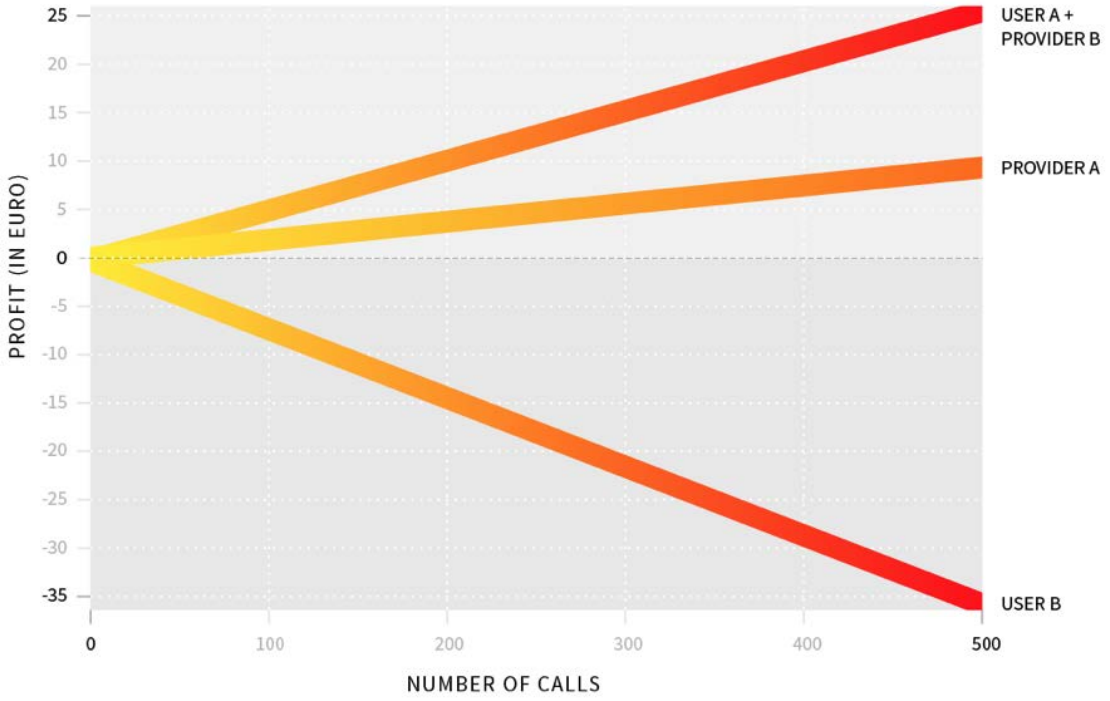


Figure 3.7.: Simple example graphic for e3Fraud tool for the Telecom case study where the TRESPASS visualisation rules have been applied to.

In the following paragraphs we describe a set of platform principles for the development of insightful and security visualisations that have impact. After describing them, we show

examples of how they can be applied to various types of graphs using various types of media.

3.5.1. Parameterisation of visual elements

Every type of visualisation or graph will contain graphic elements whose rendering is modified by variables. A circle, for example, has several variables, such as position, radius, fill and opacity, that affect the final visual outcome. One can think of these variables as controls that can be adjusted depending on a certain input. This way of thinking allows the creation of many combinations of visual elements.

For example, directed graphs[1] are used heavily in security models because of their ability to represent complex network relationships between entities. These graphs consist of edges and vertices, visually often represented by the two essential elements of a line and a circle. Feedback from early TRESPASS explorations found that it is most visually effective to parameterise a maximum of three variables simultaneously. A line can be defined, for example, by its thickness, colour, and opacity. By applying a mapping from a derived security vocabulary (e.g. difficulty, time, and probability) to the visual vocabulary, it is possible to begin building the framework for visualisation. Thickness can be mapped to indicate difficulty, colour to indicate time, and opacity to indicate probability. Note that certain properties are more suitable for mapping to certain visual parameters. It does not make sense, for example, to map time to opacity because a lower opacity implies a weak link, whereas time only indicates the length of time to complete. More examples can be found in (Bertin, 1967). All of these parameters combine to form a very rich, yet simple visualisation that leaves no visual 'stone' un-turned. A similar approach can be applied to the visualisation of information in a node.

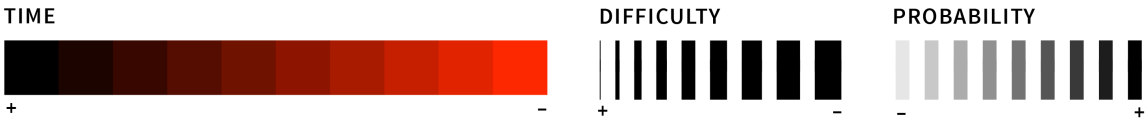


Figure 3.8.: Legend for visualising three parameters in attack trees; difficulty is indicated as stroke width; time is indicated as stroke colour; probability as stroke opacity.

3.5.2. Stacking visual elements

There are often cases where the number of parameters that need to be visualised far outnumber the sensible variants to a particular visual element. In other cases, it may not be known what the total number of parameters is (as they may shift depending on user input). An example might be an entity, such as an attacker profile, for which the user has the freedom to pick and choose the parameters to assign. In such cases, it makes

[1] Here we discuss graphs in the mathematical, not the visual context. So a graph in this context is defined as a network of vertices or nodes connected by (directed or un-directed) edges.

sense to develop a language that is extensible. An approach in which visual elements are stacked provides a solution to the problem, as it can be applied to a wide range of parameters.

First, it is important to establish a unified legend, where, for example, line thickness and colour are used to indicate levels of risk. Quite often parameters in security models can be mapped to a scale that ranges from low to high risk. This means that a visual element becomes a generalised module for visualising, allowing for adaptability and re-usability.

Figure 3.9.: The principle of stacking visualisation elements, in this example four parameters. Note that low as well as high can be represented with a thick and red line, depending on the type of parameter.

Attacker profiles are a good example of the need for stacking because the number of parameters change depending on the situation. Intel provides a good set of baseline attacker profiles in (Rosenquist, Matt and Intel IT , 2009). But there are cases where perhaps some parameters may not matter. It is necessary to create a visual system that allows for this. By using a unified legend, as described above, where thickness and colour can represent threat level, it becomes possible to represent an attacker profile as a set of stacked circles, in which each parameter is one of the circles (Fig. 3.10). This technique allows extensibility if say, later on, a situation calls for an additional parameter by providing the ability to stack an additional circle. Again, it is important to pay attention to visual hierarchy as parameters that are closer to the outside of a circle are weighted as visually more important. This can be adjusted by arranging the parameters in order of importance, or preference, from inside to outside.

3.5.3. Multiple views

Every visualisation foregrounds certain aspects of the data it is representing, while backgrounding other aspects. As not all viewers are interested in the same aspects, multiple views offer a good way to cater to this. In addition to multiple views on the same data, it is also possible that a certain model can be analysed by various tools. Each of these tools provides its own outcomes and therefore needs its own visualisation, because each view tackles a certain aspect of the security model. Viewed as a whole, often in a dashboard-like view, they paint the entire picture of the visualisation.

3.5.4. Contextual awareness and highlighting

A key aspect in security visualisation is the ability to highlight key points of vulnerability. This tends to be much more effective than just textual output, as it also gives viewers the

Figure 3.10.: Detail of visualisation of attacker profiles based on the Threat Agent profiles by Intel. Demo at: http://lustlab.net/dev/trespass/visualizations/profiles/.

ability to contextualise potential points of interest in the model. Oftentimes there are analytical tools that can provide insights such as the weakest or cheapest path in an attack tree. It may also make sense to highlight certain connections based on user interaction. It is also important to consider how and when certain elements will be highlighted when developing a vocabulary. Depending on what needs to be highlighted, certain approaches may be better than others. Contextual awareness also allows a fine-grained representation of information without overwhelming the viewer.

3.5.5. Semantic zooming

Often security visualisations will either be too simple in their attempts to abstract a model, or too complex and confusing by trying to show all the data. An approach to address this issue is *semantic zooming* which applies meaning to different zoom levels. As more and more visualisations are digital in format, taking advantage of interactivity allows one to generalise parameters of a security model depending on the level of inspection, providing more abstract representations at a macro view, and only displaying the complex intricacies when in a micro view. This corresponds to the act of zooming into an online map to reveal additional details.

This approach complements the of stacking visual elements, as it allows elements to be seen only when such detail is required. For example, when viewing attacker profiles from a macro view, it is only important to show the total perceived threat that an attacker has. As a result, the attacker can be represented by a single circle whose radius is the sum of the stacked circles. A zoom provides a more detailed view, (Fig. 3.11) where a viewer might want to inspect how parameters differ between attackers; only at this point does the visualisation reveal the individual stacked elements. By displaying this detail only when

necessary, it is possible to create visualisations that can be relatively simple without any loss of information while still allowing the viewer to take a closer look.

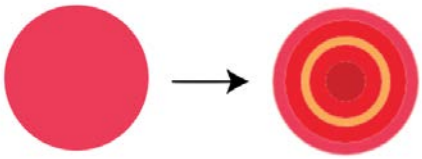


Figure 3.11.: On the left a generalised, zoomed-out state of an object, on the right a detailed view that allows for a specific inspection of each element

3.5.6. Visualising uncertainty

As mentioned, visualising uncertainty is often vital to the accuracy of a visualisation, as it allows viewers to understand the relative fuzziness of certain factors of the model. Existing work in visualising uncertainty can be found in (Harris, 1999)(Kirk, 2015). Possible approaches to visualising uncertainty include transformations such as blurring visual entities, or introducing a way of displaying multiple possible predictions of a model, similarly to how a user might choose to highlight a certain path or visual element. As an example, Harris introduces the concept of fuzzygrams, in which "columns [of a histogram] are replaced by blurred or fuzzy areas to indicate the probability of the values plotted. The degree of fuzziness is inversely proportional to the sample size." (Harris, 1999).

Figure 3.12.: Example of how uncertainty (or confidence in the data set) can be visualised.
Left: the legend
Right: application of the legend to an attack path

4. TRESPASS visualisation innovations

WP4 work on the TRESPASS project has produced a number of visualisation innovations where we have worked with the general principles and techniques described above in order to innovate aspects of security visualisations. These innovations are all available through our visualisation showcase [1]. We describe the key visualisation contributions in this chapter.

Our visualisation innovations can be summarised as follows:

- Attack Navigator Map (ANM) that provides an interface specifically designed to visualise socio-technical environments and give emphasis to socio-technical risk calculations.
- Attack tree innovations that produce linear and radial forms of attack trees to present multiple attack tree variables and to make the visual form more accessible.
- Visualisations of an attack landscape in the both the digital sphere (attack cloud) and the physical sphere (ATM case study).
- Revising the attack graph form to use the TRESPASS visualisation principles.

These innovations have both used standard interaction techniques and principles as well as innovate the visual form. For example, the attack tree visualisations have made use of combining the filter-and-focus techniques with highlighting techniques. The abstraction techniques offer a means of 'drilling-down' into individual sub-systems and these standard techniques have been used in different ways to produce novel visual forms such as linearized attack trees, attack clouds and attack landscapes.

Similarly, the ANM uses standard interaction techniques such as zooming but does so within an interface that is specifically designed for socio-technical risk analysis. Shneiderman and Plaisant (2003) describe zooming as one of the seven basic interaction tasks. In computing, a zooming user interface is a graphical environment where users can change the scale of the viewed area in order to see more detail or less, and browse through different documents. Within TRESPASS we have extended the concept of zooming to what (Cockburn, Karlson, & Bederson, 2008) refer to as *semantic zooming*: the level of detail present in the resized object is changed to fit the relevant information into the current size, instead of being a proportional view of the whole object. The presentation of data items is altered at different scale levels.

The ANM also combines these approaches to meet different TRESPASS visualisation objectives, e.g., multiple views of the system are especially helpful when selections in one

[1] www.visualisation.trespass-project.eu

view are coordinated with all other visible views to see the selected entities in the different context/perspective.

4.1. Producing an atlas and legend

All the TRESPASS visualisation innovations utilise a common visualisation atlas and a common legend structure. This atlas can be seen as a legend, a mapping out of all the possible routes that visualisations might take. The legend (Key) to a map plays an important role to help people understand what they are looking at. The legend is also one of the ways in which we communicate the TRESPASS narratives. In most cases, the design of a legend is an after-thought, an often disorganised quadrant on the bottom right-hand corner of a map or a visualisation. In our approach, however, the legend is central to the development of the visualisation tool, and is to be used as a link between concepts of information security, risk, and its modelling through visualisation. It requires the designer to establish clear goals in order to provide a clear mapping from the language of the model to the visual vocabulary. It defines which aspects of the model are represented and how they appear visually. As the legend is one of the most overlooked but important elements in a visualisation, it should be the starting point for each project; an integral part of the information presented.

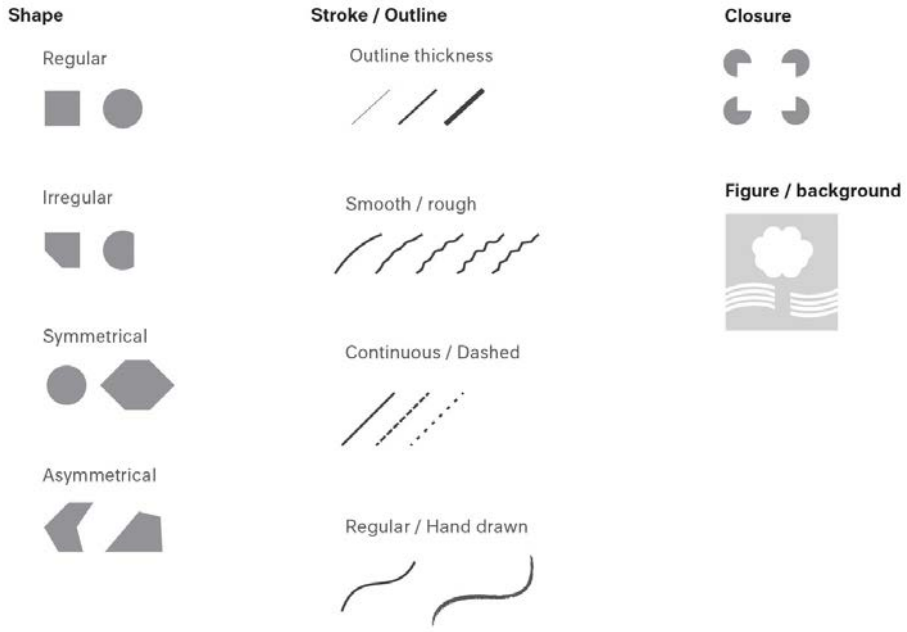


Figure 4.1.: Shape, stroke, outline. Part of the Gestalt section of the Visualisation Atlas, LUST, 2013.

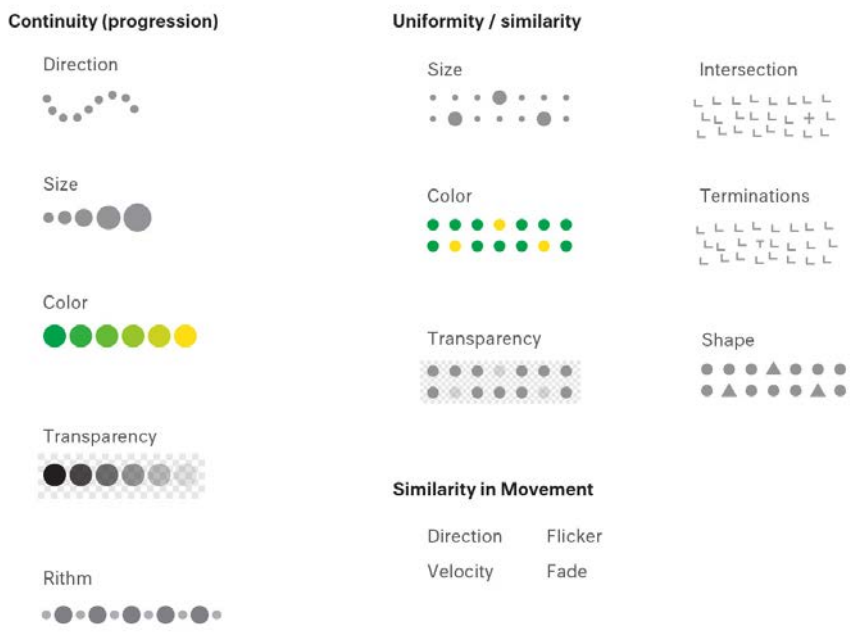


Figure 4.2.: Continuity and Uniformity. Part of the Gestalt section of the Visualisation Atlas, LUST, 2013.

4.2. Application to the ANM

The Attack Navigator (AN) is an environment where all tools developed within the TRESPASS-project can be viewed, accessed, used and connected. The requirements, prototype specifications, and how they are interpreted can be found in (The TRESPASS Project, D6.3.1, 2015). A main part of the AN is one of the major tools, the Attack Navigator Map (ANM).

The design brief for the TRESPASS user interface was based on the functional requirements laid out in various deliverables (The TRESPASS Project, D6.1.1, 2013); (The TRESPASS Project, D6.2.2, 2015). In addition to these formal requirements there are also other notions that have been formed through discussions, meetings and brainstorming sessions.

4.2.1. ANM design concept and structure

We address here only the Attack Navigator Map from a visualisation standpoint, for the Attack Navigator please refer to (The TRESPASS Project, D6.3.1, 2015). The primary purpose of the ANM is to describe a landscape of potential risk and to provide visual tools to explore that landscape using the underlying risk calculation algorithms provided from the work in WP3.

As the structure of elements in an Attack Navigator Map can become complicated very quickly, a wizard-like structure is applied, that guides users through the various steps that need to be taken. Users can draw or import floor plans (for physical and digital environments), apply those to multiple floors and drag-and-drop items (Fig. 4.3) as assets and actors onto the map. These assets, actors, and many more items come from libraries, where the user can also save their own library items, add items, and adjust the properties. In (The TRESPASS Project, D6.3.1, 2015) it is described as follows:

The basic building blocks for constructing a model come from libraries of single components, or from prefabricated model fragments (groups of components with relations), such as the model pattern library. These libraries will contain commonly used patterns, that can be used as templates to rapidly build the basic structure, which can then be refined and tweaked. The underlying data structure is a directed graph of nodes (components with properties) and edges (relations between those components).

4.2.2. Rationale for developing a new visual editor

The Attack Navigator Map tool is implemented as a single-page web application. We deliberately chose to develop a web-based solution for a number of reasons. Importantly, the ANM's sister tool, the Attack Navigator (AN) is already web-based, and in order to share code between them, we needed to develop a visual editor that could do so. Furthermore all of TRESPASS' tools are designed with the philosophy of embracing the fact that the web is essentially platform-independent. Such an approach also enables us to

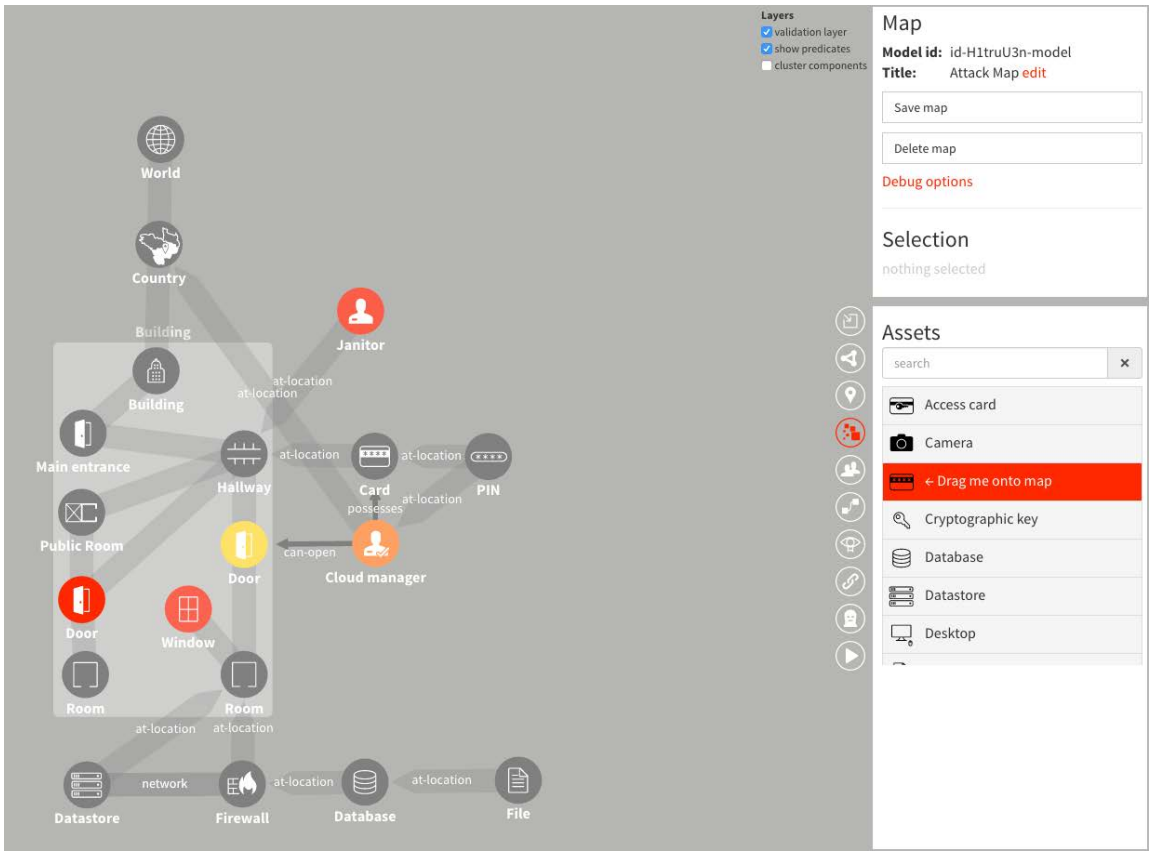


Figure 4.3.: A map created in the ANM. The user hovers over the asset "access card" and is prompted that the item can be dragged onto the map. The colour of assets is based on how potentially dangerous the asset is. For "door" red means very weak, for an actor type red means vulnerable.

distribute the ANM as a standalone web application, this would be possible without much extra effort, using technologies such as Electron[2].

One of the main features of the ANM is model editing, in what TRESPASS terms the "map editor", which is based on the idea of expressing a socio-technical model through nodes of certain types and connections between them, which describe their relations to one another. In order to implement this graph editing capability, we initially looked for existing solutions, and evaluated whether the existing solutions were suitable to express the outputs of TRESPASS analysis.

We evaluated off-the-shelf tools against the following criteria:

- The tool had to be freely available.
- The tool had to have its source code fully available.
- The licensing of the tool had to permit modification / customisation.

[2] http://electron.atom.io/

The customisation aspect is especially important with regards to the following two points. We needed full control over the visual appearance of the output, in order to make it conform to the visual language we developed previously as part of our visualisation work. The Navigator Map is not only a visual representation of a graph structure, but also visualises properties of the components on the map as an indicator for potential vulnerabilities, for instance. It also needed to be customisable so that we could ensure that the visual editor was fully compatible with the underlying data structure of the TRE$_S$PASS socio-technical model format.

Given these criteria, all commercial components and libraries were not an option. Of the remaining ones many were merely "diagramming" tools, that support drawing of graphs, but could not export of structured information from it.

Our evaluations revealed that the most likely candidate was the open-source project The-Graph[3], which is developed as part of the Noflo[4] project — a flow-based / visual programming platform for javascript.Therefore, for one of the very first ANM prototypes we used The-Graph, for the purpose of evaluating its potential.

Our evaluation showed that The-Graph had the following advantages:

- Open source
- Uses modern web technologies (react, SVG)
- Has many helpful features built-in (zoom / pan, grouping, autolayout)

But there were also disadvantages:

- Has unnecessary / outdated dependencies: (an old version of) Polymer[5]
- Is designed to be used with horizontally aligned graphs (which is not how TRE$_S$PASS models work)
- Though somewhat actively developed, it was lacking proper documentation for developers (which would have made it hard to get familiar with the code in reasonable time)

Given the disadvantages, we decided to develop our own editor but we based some of the design on the relevant The-Graph features, and in some parts were even able to re-use parts of the original code.

In our eyes this was the better option, not only in terms of entirely controlling the visual output, but also with regards to catering to the specifics of the TRE$_S$PASS socio-technical model specification, and the bigger task of creating an meta-tool that integrates model creation, analysis, and analysis results visualisation in one user interface. Although the map / editor view may be the most prominent one, the ANM does much more then just “draw maps”.

[3] https://github.com/the-grid/the-graph
[4] http://noflojs.org/
[5] https://www.polymer-project.org

4.2.3. ANM analysis results dashboard

The Attack Navigator Map tool unites a large part of the TRESPASS tool chain (model creation, attack tree creation, analysis, visualisation) in one user interface. The analysis results visualisation dashboard is the last step in the tool chain, and will appear as a different view on top of the regular ANM user interface. It gathers all the results of the analysis (and other intermediate tools) and makes them available to download, and visualises them as attack trees. Next to this the dashboard also offers alternative visualisations, that are derived from the attack tree. If needed it also displays additional visualisations, that are specific for the output format of individual tools, for instance the Attack Cloud visualisation (See section 4.5) and the tree map view (Fig. 4.6).

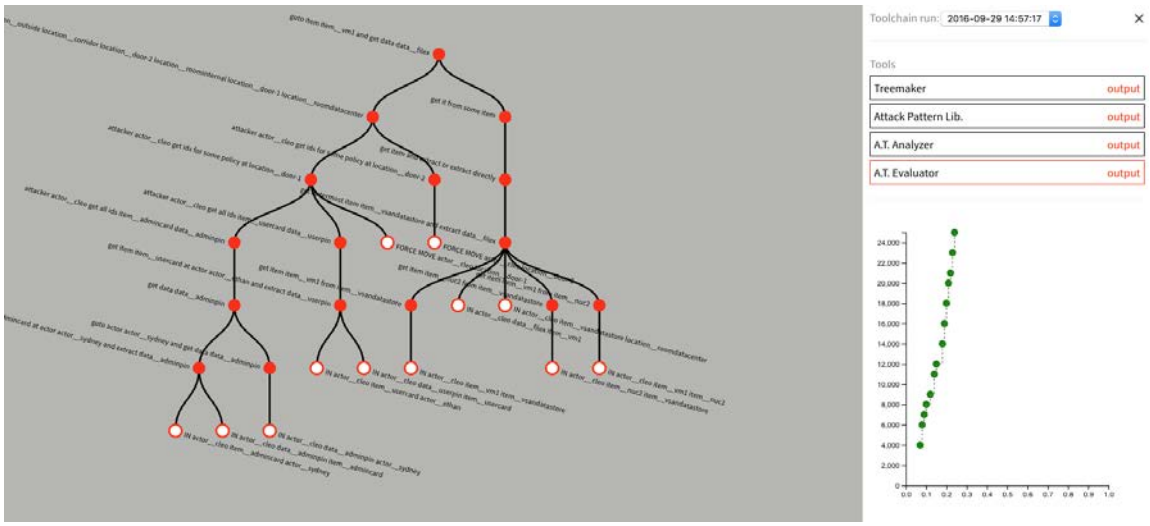


Figure 4.4.: The panel on the right shows the list of tools that have run. Each of them is select-able, and when selected the respective data is visualised on the left. In the case of the analysis tools, additional visualisations are displayed below the tools list. Selecting a specific result / attack will also update the main visualisation on the left. Hovering over one of the green nodes in the pareto frontier on the right will highlight nodes or edges in the attack tree in the left.

4.2.4. Integrated and stand-alone

The TRESPASS visualisations are developed as single, loosely coupled components – entirely independent of one another. By avoiding interdependence we can ensure complete modularity, which in turn allows us to use the components as building blocks for applications like the ANM analysis results dashboard, with the option to easily replace components with compatible alternatives, if needed).

At the same time it is also entirely possible to take a single visualisation component, and –with only a thin layer of application logic around it– package it and distribute it as a standalone (desktop) application.

How it works The javascript framework that is used is react[6], where components take required or optional inputs (very similar to function arguments in programming), called "props" (short for properties). The input data must be provided in a certain format, which can be a common format shared among similar visualisation types (attack trees, for instance), or specific to the output of individual analysis tools (for instance ATEvaluator). Best practices dictate to build components that only contain a minimum amount of logic themselves. The task(s) of parsing and preparing the input data is therefore handed over to the host application, whose responsibility it is to provide the component with the right data.

All of the preparation and pre-processing routines are outsourced into an external library, and available as reusable utility functions. The trespass.js library has sub modules for working with the TRE$_S$PASS socio-technical model format, different "flavours" of attack trees, and the output formats of the analysis tools that are part of the TRE$_S$PASS family of tools.

4.3. Application to attack trees

Attack trees, as developed and used in the TRE$_S$PASS consortium, are a tool to capture all possible attacks to reach a specific goal, as described in the root node. To build such an attack tree, experts typically gather and, starting from the goal node, try to enumerate possible ways of attack to reach this goal. Each sub node can be iteratively refined as far as it seems fit. Individual intermediate nodes can thereby be either conjunctive or disjunctive. A conjunctive node requires that all of its children be fulfilled in order to proceed up the tree, whereas disjunctive nodes only require one child to be satisfied in order to proceed. A complete path of actions consists of any number of leaf and intermediate nodes leading to the root node. Within the project, each leaf node is considered to contain four parameters: difficulty, minimum cost, probability of success, and minimum time required to complete. Depending on the effort put into the creation, these attack trees can be very complex, comprising hundreds or thousands of nodes, especially when they are generated programmatic-ally from an underlying model as is the aim of the project. When attempting to visualise these trees, such visualisations quickly become complex and unreadable.

In their traditional form, attack trees present a wide variety of important and relevant information, but are not easily visualised, oftentimes shown as an arrangement of text in a directed graph. From a visualisation perspective, attack trees have several flaws; the tree structure gets very wide rapidly, repeating lots of elements to eventually become effectively unreadable even in a medium allowing arbitrary zooming. Also, because attack trees consist of conjunctive and disjunctive nodes, it needs to become visually clear that in the case of conjunctive nodes, all steps need to be fulfilled in order to proceed. We can counteract this complexity by improving the way the tree is laid out and labelled, as well as by testing alternative layouts that result in more compact trees, while maintaining

[6] https://facebook.github.io/react/

readability. Next to that, exploring interactivity by allowing the user to zoom and pan, and to collapse sub-trees at any level, makes it easier to concentrate only on certain parts of the tree.

The key components and their respective properties are the following:

Node	**Edge**
Type of node (leaf, intermediate, root)	Parent/Child nodes
Conjunctive or disjunctive node (if intermediate or root node)	
Label	
Minimum cost to complete node	
Probability of success	
Difficulty	
Minimum time required to complete	

A visual language can be developed based on this. It is important to keep in mind that attack trees can vary greatly in size, as their construction is largely dependent on the scenario and environment that they are trying to model. As a result, the language should be scaleable to any size tree. In initial explorations, user feedback revealed that when representing the graph with directed graphs, edges carry much more weight and information visually. Subsequently, most of the parameters were mapped to the edge leading to parent nodes. This allows focus to be placed on the path rather than on each individual step. The resulting legend was developed by parameterising the visual properties of a line (Fig. 3.8) and creating a mapping to the attack tree vocabulary (Fig. 4.5).

Multiple views Visualising an attack tree in a tree structure may do a good job at displaying how the nodes are connected, but it does a poor job of examining frequency. Therefore, it makes sense to split this into two visualisations: an attack tree visualisation structured as a tree, as well as a tree map visualisation that focuses just on the relative frequency of each node (Fig. 4.6). The frequency of the node determines the size of each box, while the colour depicts the relative difficulty of each node. A hover over each box in the tree map shows its label and highlights the nodes in the attack tree, allowing a user to understand the visualisation. Together, they paint a more complete picture. We consider the tree map as part visualisation and part legend.

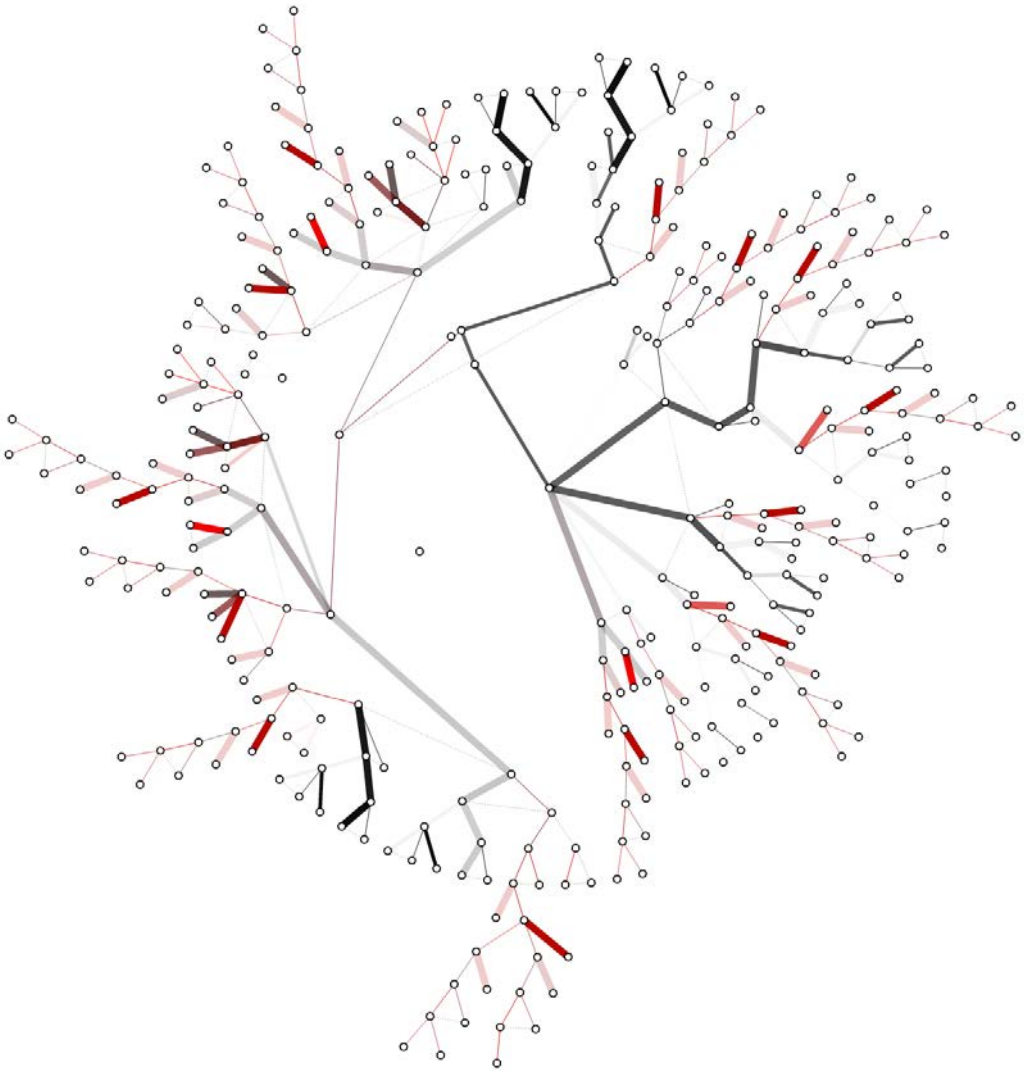


Figure 4.5.: Attack tree visualised in radial form where each node corresponds to an attack step. The root node, i.e., the goal, is placed in the center. The edges are coloured according to three parameters: difficulty is indicated as stroke width, time as stroke colour, and probability as stroke opacity.

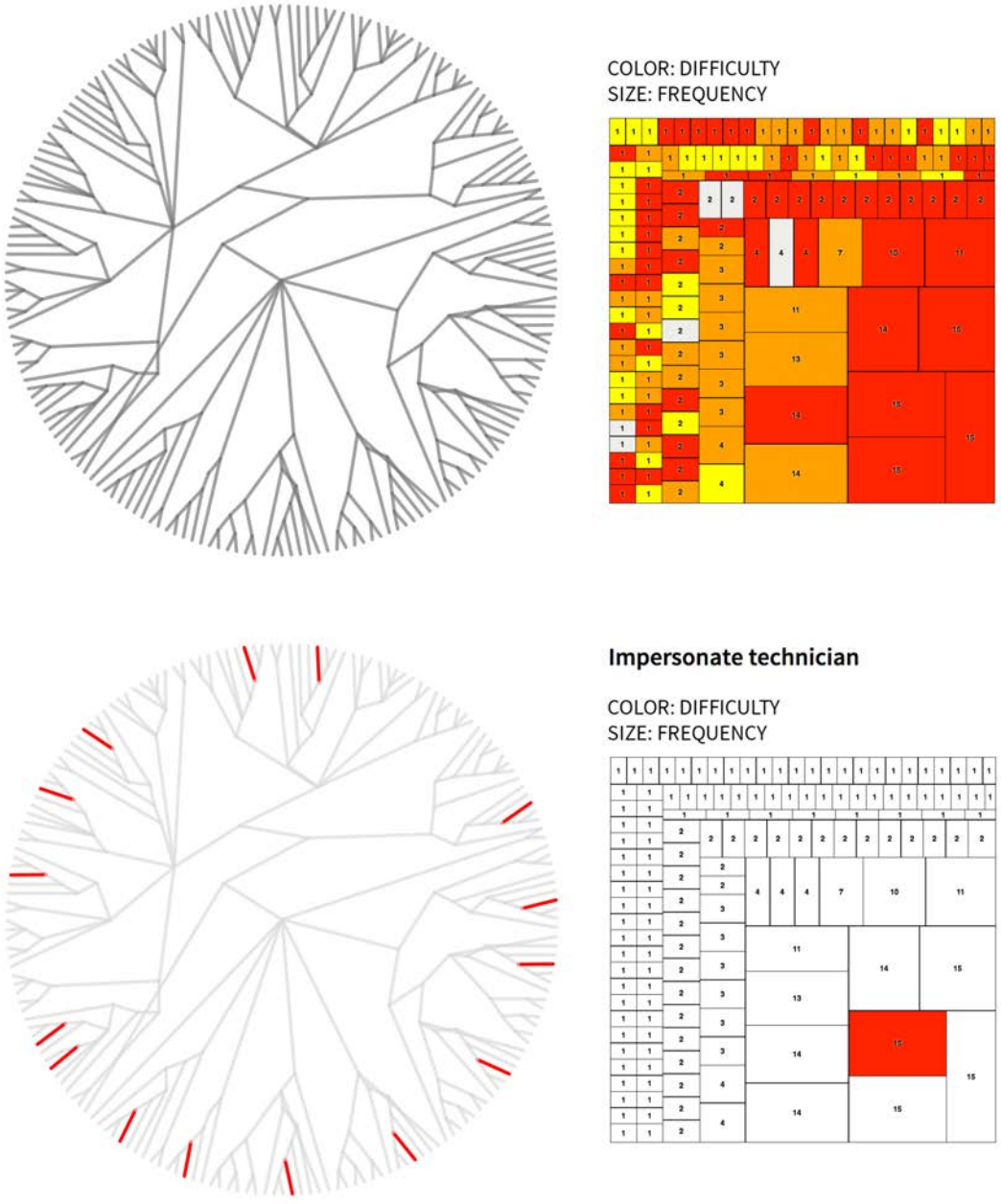


Figure 4.6.: Top: Tree map visualisation that shows frequency of an attack step.
Bottom: figure shows a user hovering over the treemap, highlighting the "impersonate technician" box. All repeating instances of this node are highlighted in the attack tree visualisation

4.3.1. Attack Tree Linearisation

In visualisations, it is widely agreed that it is better to have more simple elements than fewer, complex elements (eg. (Tufte, 1990)). A tree works well in situations where the structure is fairly simple and small. However, the attack trees that are used in TRE$_S$PASS are already more complex than can comfortably be fit on a screen. Working with and studying attack trees from a visualisation point of view, one can question the role of intermediate nodes. Other than being a labelled container for their child nodes, they are not actually steps along the attack path but nevertheless occupy a large part of the attack tree. We can visually simplify attack trees by turning them into linear sequences of their required children. This will result in more paths, but each path will be easier to follow. The simplification and conversion to straight paths benefit readability from a visualisation standpoint. One path now shows a user the steps that need to be taken in a straight and easy to follow line (although it does not usually imply a temporal or causal sequence) (see Figures 4.7–4.12).

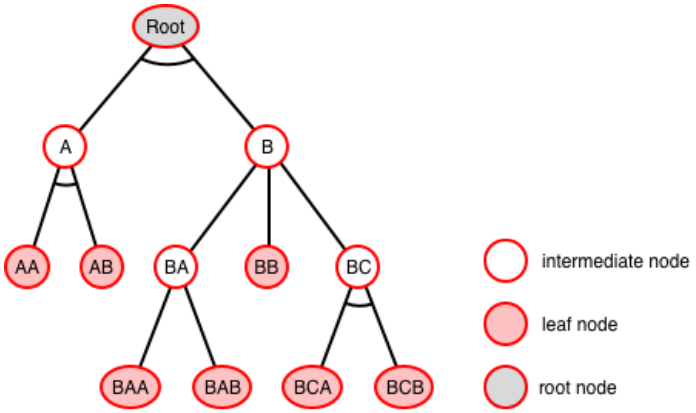


Figure 4.7.: Example Input attack tree

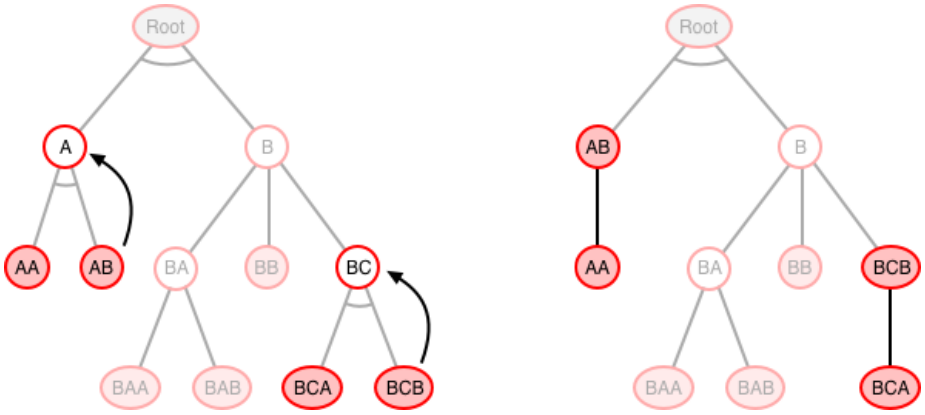


Figure 4.8.: Left: In the first step the algorithm finds all conjunctive intermediate nodes. It traverses the tree depth-first, thus processing A and BC before Root, in later iterations.
Right: The conjunctive nodes are eliminated by replacing them with a linearised form of its children. Each sibling becomes the child of its right-hand neighbour. In this example all siblings are leaf nodes, resulting in a linear chain.

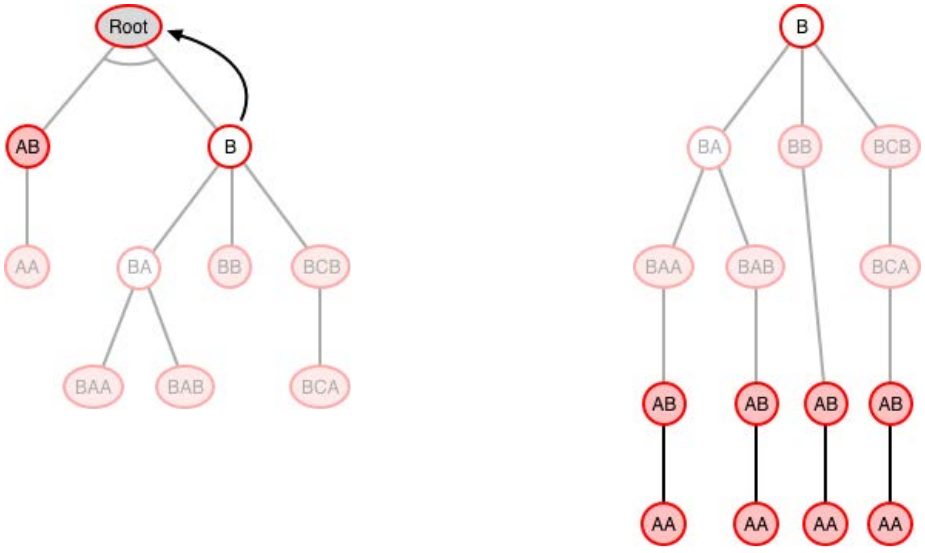


Figure 4.9.: Left: In the case of the Root node, the siblings (AB and B) are subtrees, rather than single nodes.
Right: Instead of becoming direct children of the next sibling, each child's subtree gets attached to the leaf nodes of the next sibling's subtree.

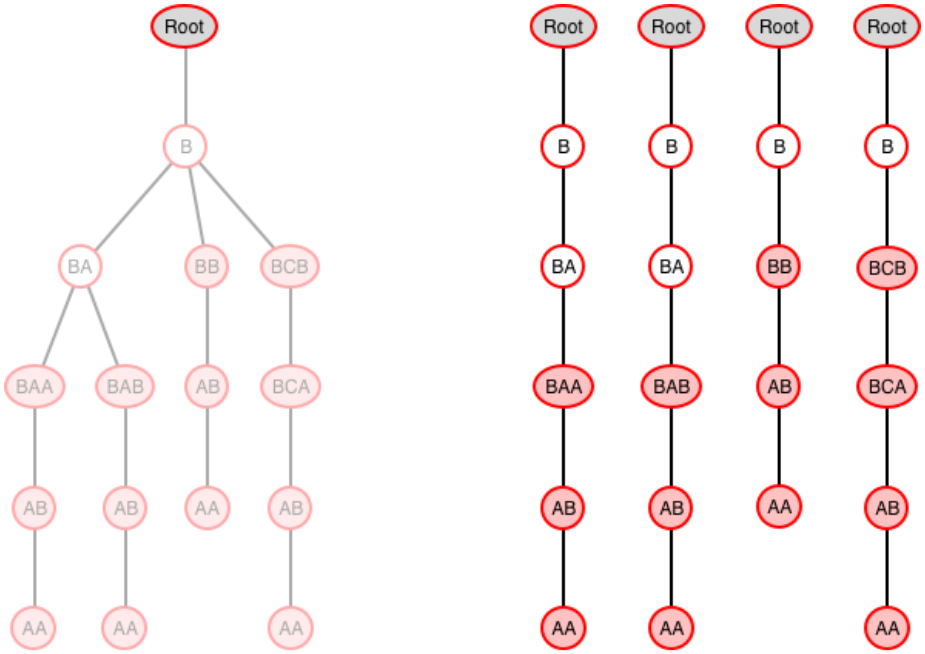


Figure 4.10.: Left: Since we want to keep the goal of the attack tree, we need to add the root node again.
Right: In the second step, we extract all the individual paths from the transformed tree

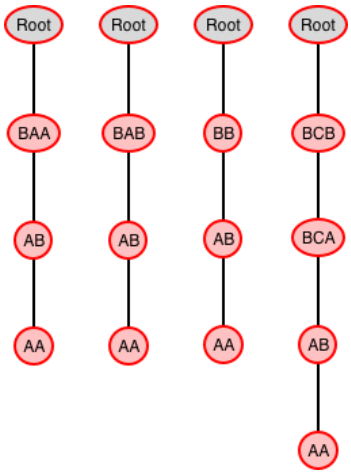


Figure 4.11.: Finally we remove all remaining (formerly) intermediate nodes.

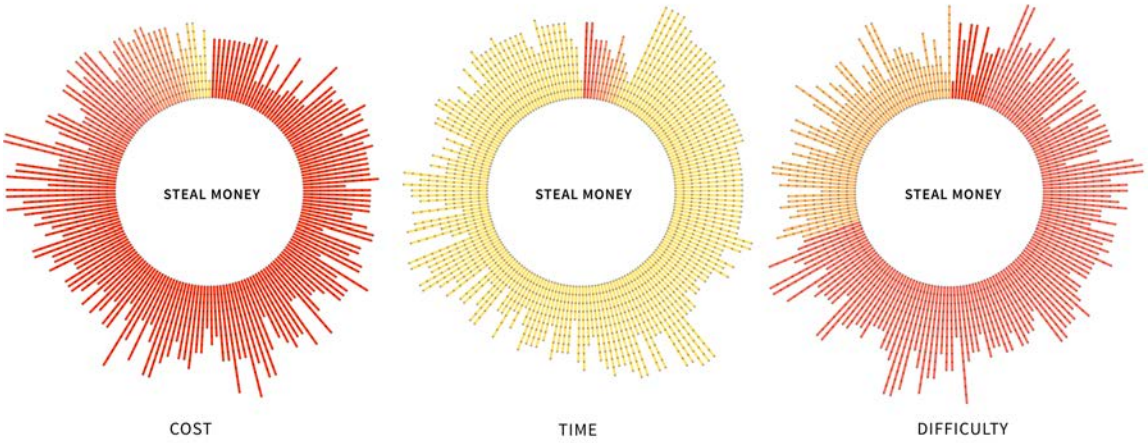


Figure 4.12.: Three visualisations of the same attack tree as linear paths for each attack trace, each depicting one parameter (cost, time, difficulty). All values of each attack trace are generalised to one total value. They are ordered clockwise where the top is the most vulnerable attack trace

4.3.2. Stacking Visual Elements

Another legend was also developed for the scenario when additional parameters to each step may be needed. By mapping different visual elements (thickness and colour to threat level) of a line to a scale of threat, it is possible to modularise this element and stack it to any number of parameters.

Visually, this becomes just as effective as the original legend because a step in which all parameters have a high perceived threat level will stand out much more strongly than a step with a low perceived threat level. When combined to form a path, as in Fig. 4.13, this legend is very informative on which steps and connections are areas of vulnerability.

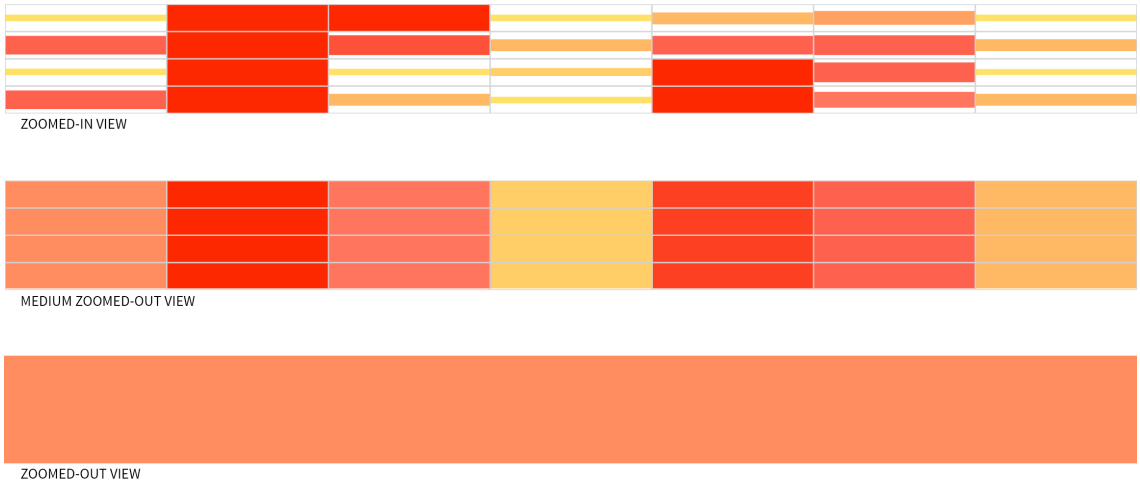


Figure 4.13.: Top: application of stacking elements to an attack trace in an attack tree. Each parameter has its own space and can be inspected individually. Generally this would be used in a zoomed in detail view. Middle: application of stacking elements to an attack trace in an attack tree. Each node is generalised to the average parameter values. Bottom: Most general view of the attack trace, where the values of each parameter are abstracted to one value.

Figure 4.14.: Alternative view on the same data as Figure 4.13, where the visual elements are stacked without inbetween space, therefor adding total height as a visual clue for each node. Also this view can be abstracted depending on zoom-level.

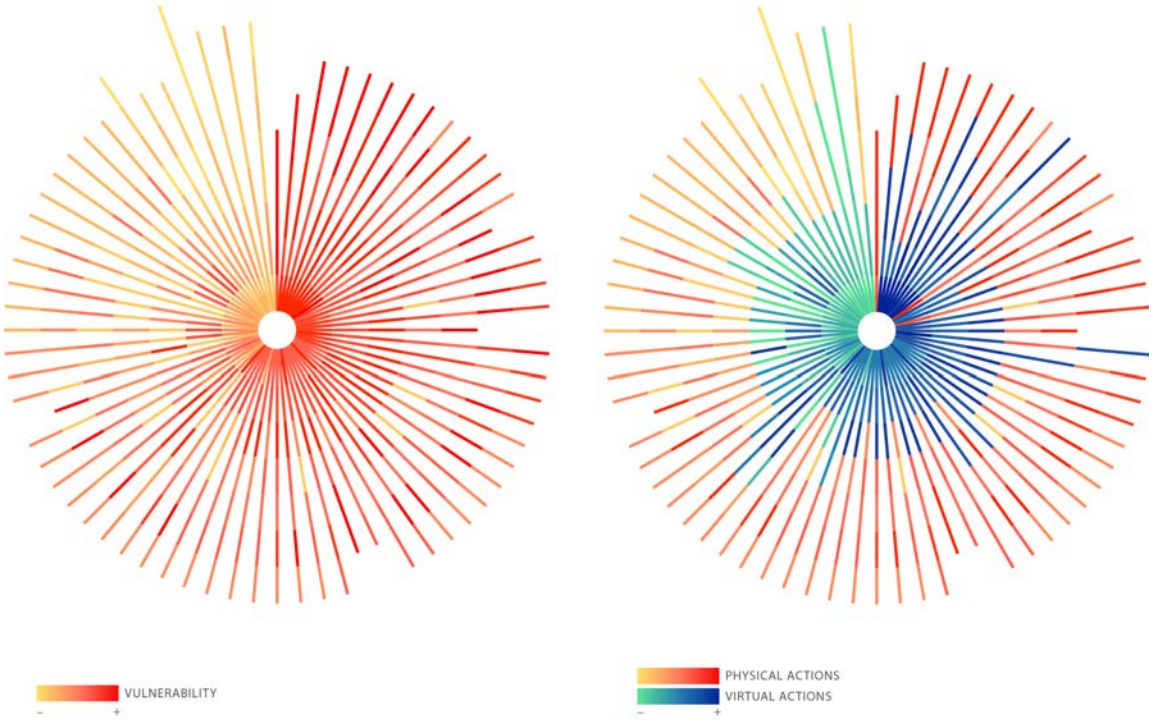


Figure 4.15.: Two visualisations of the same attack tree visualised as attack steps on attack traces, both ordered clockwise where the top is the most vulnerable attack trace. On the left, only vulnerabilities are highlighted, while on the right a differentiation is made between physical nodes and virtual nodes.

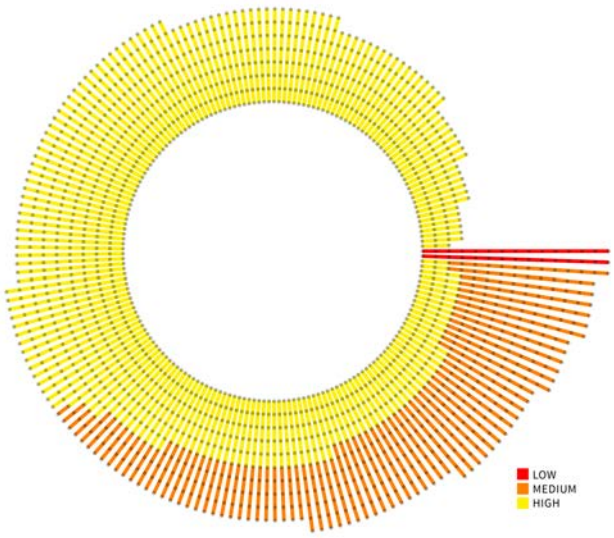


Figure 4.16.: Visualising attacker skills. Red indicates how far an low-skilled attacker could come, orange a medium-skilled attacker and yellow a high-skilled attacker.

4.3.3. Semantic Zooming

When applying the stackable legend developed in section 4.3.2 to massive attack trees, the overall visual effect of the visualisation becomes confusing and harder to read. Viewers are not as able to follow paths as easily as before. However, semantic zooming can be applied in multiple ways. Because the stacked lines are only necessary at a very detailed level, it is perfectly fine to show the average threat level at a macro view with other paths or the entire tree (Fig. 4.13). Only when zooming in to view specific paths will the individual stacked lines be revealed to the viewer. This eliminates the original complexity at a macro level while still allowing specificity at a micro level.

This can be combined with a rearrangement of the linearised attack trees to present the paths in a more understandable manner. By using a radial view for the linearised attack trees at a macro view and transitioning to a table, in which information about the total path can be displayed alongside each path upon user interaction, it can be possible to sort and analyse paths in a way that might otherwise be unwieldy at the macro level (Fig. 4.13). A viewer can then zoom in even closer to see an individual path and its stacked line components, as well as any intermediate labels that might not have been shown before.

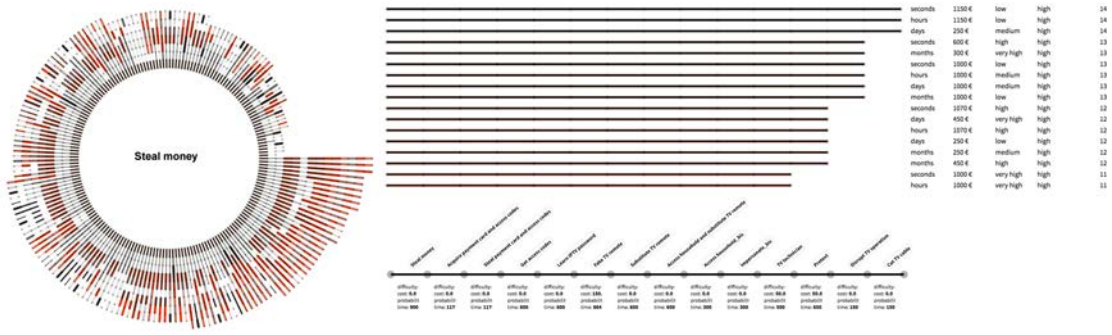


Figure 4.17.: Semantic zooming applied to linearised attack paths. From left to right: an attack tree in radial and straight path form, and the same paths as a table allowing detailed inspection

4.3.4. Horizontal attack-defence trees for the ATM case study

Most attack trees used in the TRE$_S$PASS-project only include attacks, decorated with values for difficulty or probability. The ATM case study –in addition to parameters for difficulty and probability– also implements countermeasures. As previous attack tree visualisations did not account for this yet, a specific visualisation was devised. In Fig. 4.18, the arrangement of the tree is based on a hierarchical grid to simplify the relation between the nodes and avoid repetition. The horizontal layout of the attack-defence tree facilitates the exploration of the complex data set and allows the intersection of multiple parameters. The 'difficulty' value of each node is represented with a specific colour palette from yellow to

red (high—low). The ‘probability’ value of each edge is represented with the thickness of the line (i.e. a thin line shows a low probability).

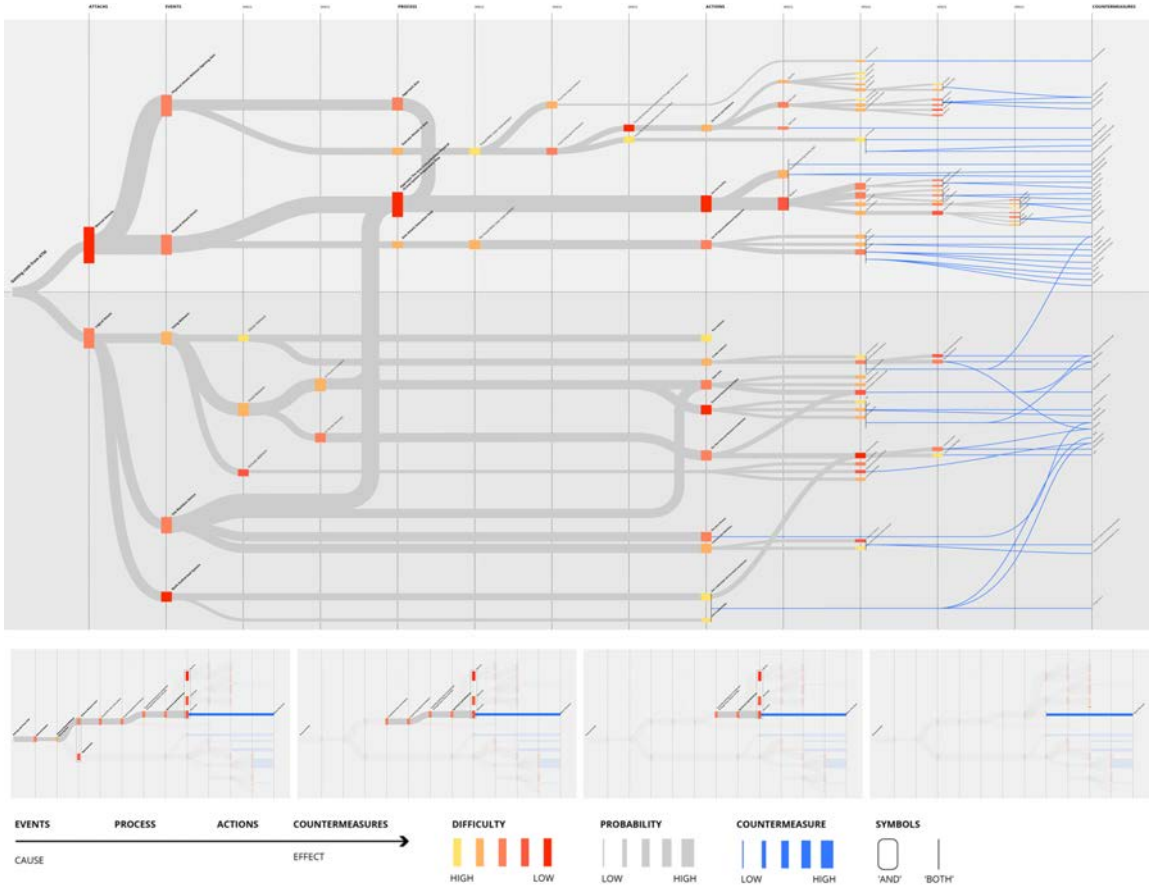


Figure 4.18.: Top: overview of all the nodes and edges in the ATM attack-defence tree, including countermeasures (in blue). In an interactive application, this view would actually never been seen as the user would be presented with simplified views for faster comprehension.
Bottom: four steps in one attack path, as user hovers over the various nodes.

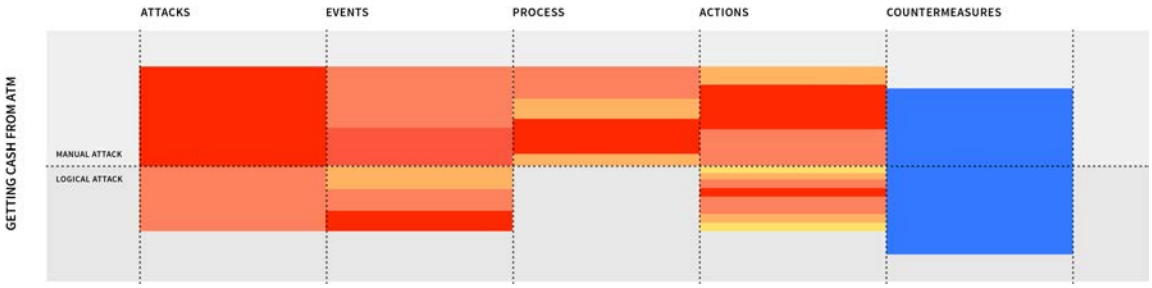


Figure 4.19.: The visualisation strategy of stacking visual elements to communicate multiple parameters, as described in section 3.5.2, is applied here to the ATM case study. The visualisation is divided into manual and logical attacks. The nodes are combined in broader attack steps as attacks, events, process, action, and countermeasures.

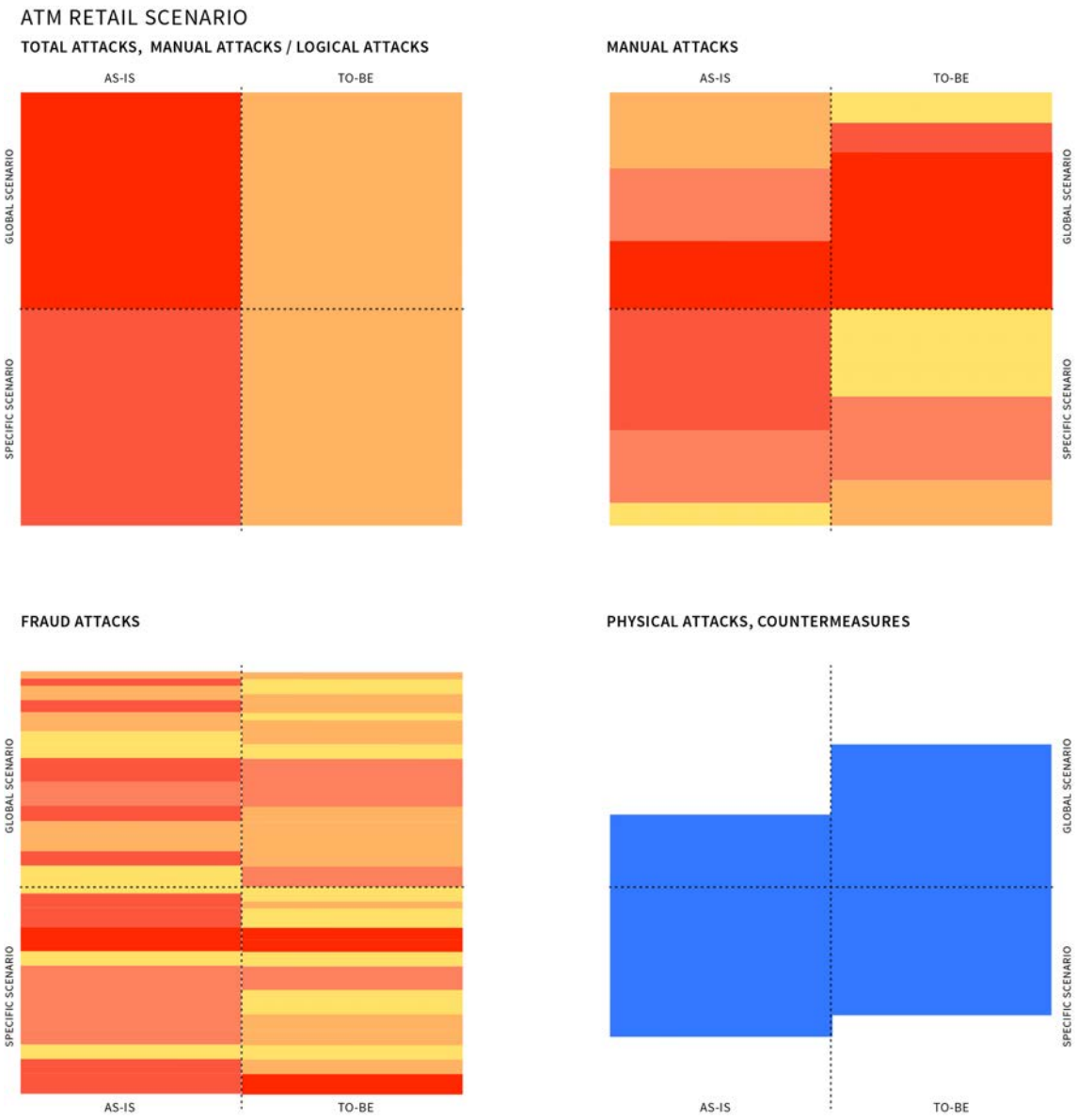


Figure 4.20.: High-level overview based on attack trees for an ATM retail scenario displaying AS-IS versus TO-BE scenarios. In the TO-BE scenarios, various countermeasures have been applied to get to the desired state of security. Top left: global overview. Top right: overview for only manual attacks. Bottom left: overview for only fraud attack. Bottom right: overview of countermeasures on physical attacks. See visualisation.trespass-project.eu for complete documentation

4.4. Applying the principles to the analysis tools

Attack trees are a widely used graphical tool for modelling the security threats of an organisation and representing attack scenarios in an intuitive manner. The root of a tree represents the main goal of an attacker, and the leaves correspond to an attacker's basic actions. The following outlines the extensions to the analytical capabilities that TRE$_S$PASS has made to the attack tree form and presents the corresponding visualisation research and innovation to articulate these extensions. WP4 has developed visualisations for the following analysis tools.

ATEvaluator ATEvaluator calculates pareto efficient solutions for the attack tree. Standard attack trees combine basic actions either conjunctively or disjunctively, thereby limiting their expressiveness. Most analyses of attack trees consider attack tree with one parameter and optimise one aspect of an attack scenario, such as feasibility or cost of an attack. Moreover, in most attack tree models with multiple parameters values, characterising basic attacks, are propagating to the root based on the local decision strategies. In case of incomparable values, this approach may yield sup-optimal results. ATEvaluator responds to this weakness by calculating pareto efficient solutions for the attack tree using two values rather than one.

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

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

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

ATAnalyzer ATAnalyzer performs quantitative attack tree analysis. The type of analysis and the outcome depends on the chosen model. Currently the two models are supported: the failure-free model and the parallel model. If the failure-free analysis is launched, the outcome is a binary value which gives an answer to the question whether the considered infrastructure is a fruitful target for rational profit oriented attackers. If the system

is analysed by the parallel model, the result is the most profitable attack vector (if any). The analysis can be done taking attacker profiles into account, as well as without profiling considerations.

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

ATtop ATtop uses priced timed automata and Uppaal SMC as the model checker to obtain quantitative values. It performs timed analysis on attack trees. It can answer stochastic and optimal questions.

The tool requires as input:

1. System model- attack tree
2. Security question translated as metric
3. Data values for basic attack steps: For stochastic questions, the tool requires mean time of execution of step and a probability of success. For optimal questions, it requires input over which optimality is desired such as costs structure to obtain minimum cost to reach the goal.

Stochastic type of questions are: What is the probability of eventually success of an attacker? What is the mean time to attack the system? What is the probability that attacker can penetrate the system in less than 1 day? How much time does the attacker need to succeed with a given percentage?

Output to these questions: a probability value at a time t or a probability value at a time $t = infinity$. This can be also expressed as cumulative distribution function.

Optimal questions determine the optimal attack values (such as minimum time to reach the goal, minimum cost to reach the goal, trade-off between attack values. Optimal questions are: Given an attacker budget, skill levels, what is the optimal cost to reach goal? Which attack path an attacker should follow, if he wants to reach the goal in minimum time? What is the maximum damage in terms of monetary loss that is inflicted on an enterprise due to attacker action/ execution of basic attack steps?

Output to these questions is a single value that corresponds to minimum time, minimum costs for an attacker or maximum damage to an enterprise. The tool can also provide an attack trace, a set of basic steps which were involved in the computation of a metric.

4.4.1. Visualisation explorations of analysis tool

Most of the analysis tools provide outcomes comparable to a 'top 10', but all do that in slightly different ways. The visualisation of the outcomes of these tools are presented with small charts on the left and a sub set of the attack tree on which the analysis is applied to on the right, and they are always linked to each other. This makes it easy for a user to look for the most vulnerable attack traces.

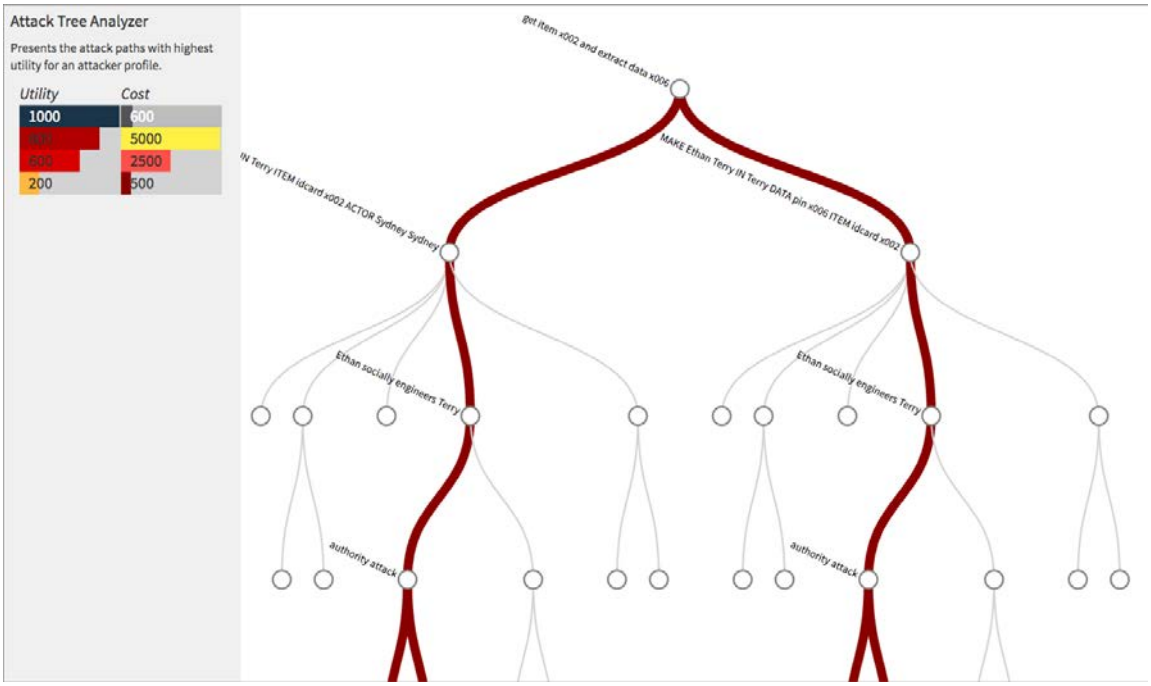


Figure 4.21.: ATAnalyzer presents the attack traces with the highest utility for an attacker. In this example a user hovers over the highest utility (utility=1000, cost=600).

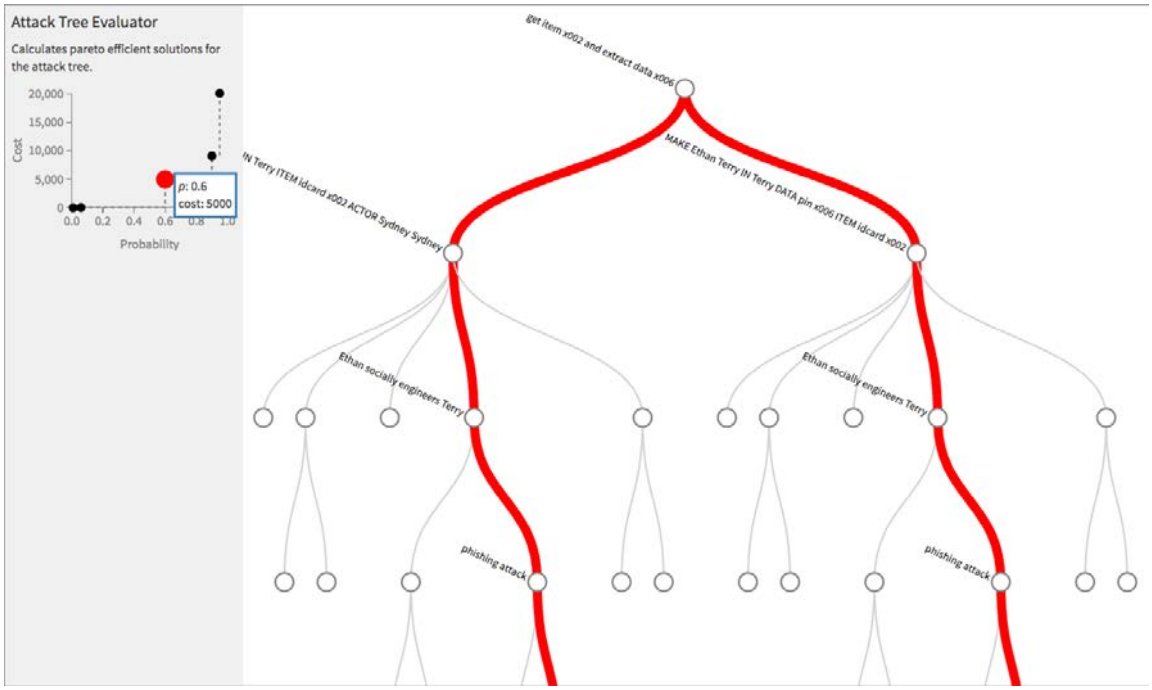


Figure 4.22.: ATEvaluator calculates pareto efficient solutions for the attack tree. Hovering over the pareto frontier highlights the involved attack traces in the sub set of the attack tree on the right.

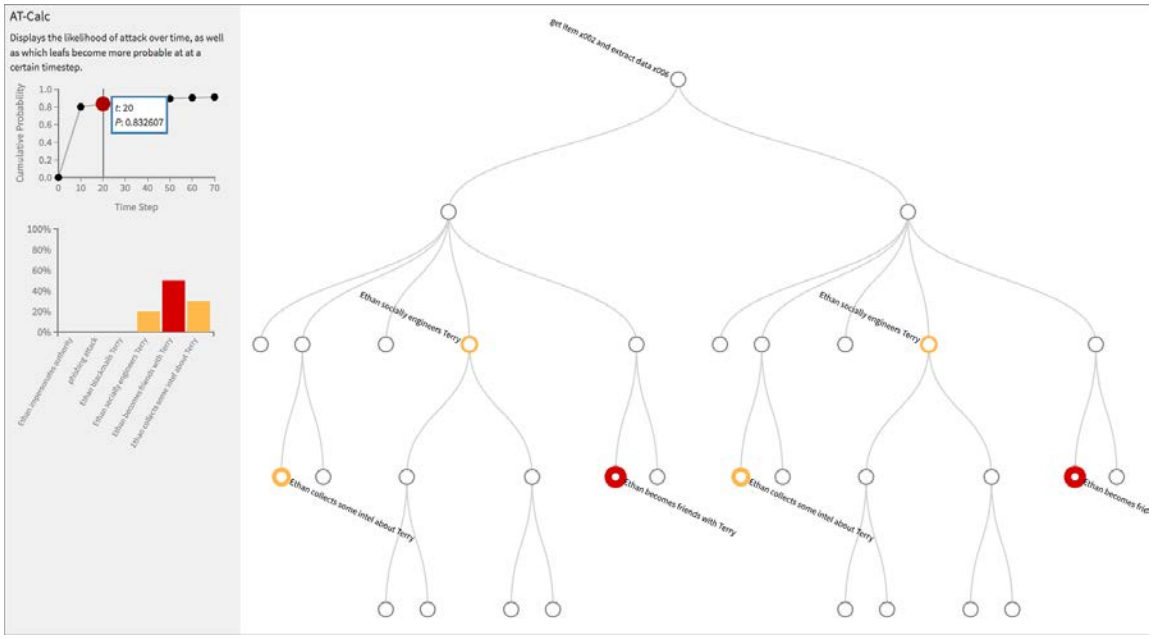


Figure 4.23.: ATCalc displays the likelihood of attack over time, as well as which leafs become more probable at a certain point in time. The two small graphs on the left plus the sub set of the attack tree on the right interact with each other so that a user can quickly explore the results of the analysis tool.

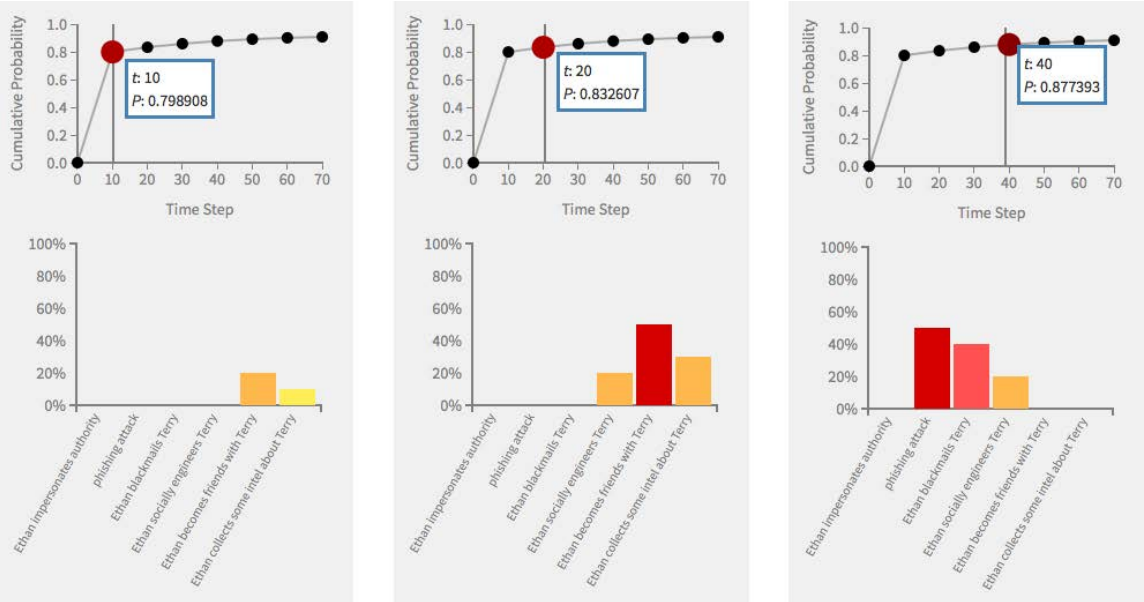


Figure 4.24.: Detail of the two parts of the visualisation of the ATCalc results. Each time step allows explorations and visualises in the graph under it which leaf nodes are involved.

4.5. Application to Attack Cloud

Representing the hierarchical nature of a structure in a tree structure or tree diagram is very common, but it also has its disadvantages. Especially in larger structures (200+ nodes) the tree form is not always the most optimal way to present a structure in a graphical form, let alone make this actionable. In an attempt to provide a better overview for very large attack trees (1.000—500.000 nodes) we developed what we refer to as the attack cloud. An attack cloud aims to represent all the steps possible in an attack tree. Because there is often no sense of order in an attack path, linearisation can potentially be misleading. This cloud format allows to see which steps are involved in which attacks while still understanding the full context. Steps that are a higher potential threat are closer to the root node at the center, which creates a logical hierarchy of information. By removing duplicates, this approach could potentially also allow us to view entire attack trees as a threat landscape.

The attack cloud graph (Fig. 4.27) is divided in sections on the basis of the main action from a label (for instance In, Make, or Force), giving a user a general idea of the action attached to a node. The placement and colour of a node are based on the combination of all known parameters for a node (for instance cost, time, probability and difficulty). The size of the node represents the number of occurrences of this node. Hovering over a node lets the user inspect all parameters, view the label with actions, actors and types, and lets the user mitigate the threat-level of a node by means of altering the parameters as counter-measures. This can be fed back into the Attack Navigator Map from which another analysis can be run.

The XML-output from the TreeMaker tool generates labels that are very long and not very 'human readable'. A typical label output would be:

```
MAKE actor__Cleo actor__Mr_Big IN actor__Mr_Big data__adminpin
actor__Sydney
```

By restructuring the text into understandable pieces, those labels can become much more informative. In Figure 4.25 this typographic approach is shown.

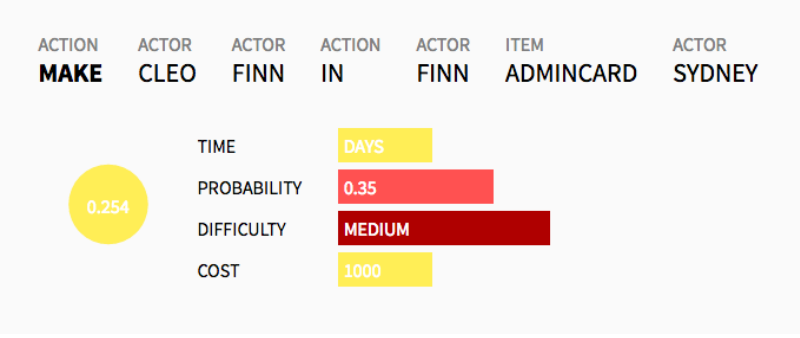


Figure 4.25.: Example of a label that appears when a user hovers over a node.

The top row consists of the various types, such as action or actor. The second row shows the type of action or actor. Below the text, a mini-visualisation of all the data that determines the position and colour of the node, is displayed. A circle indicates in colour and number the general threat score of the node in question, and a bar graph with the four parameters this general threat score is composed of.

Figure 4.26.: The legend to the Attack Cloud visualisation.

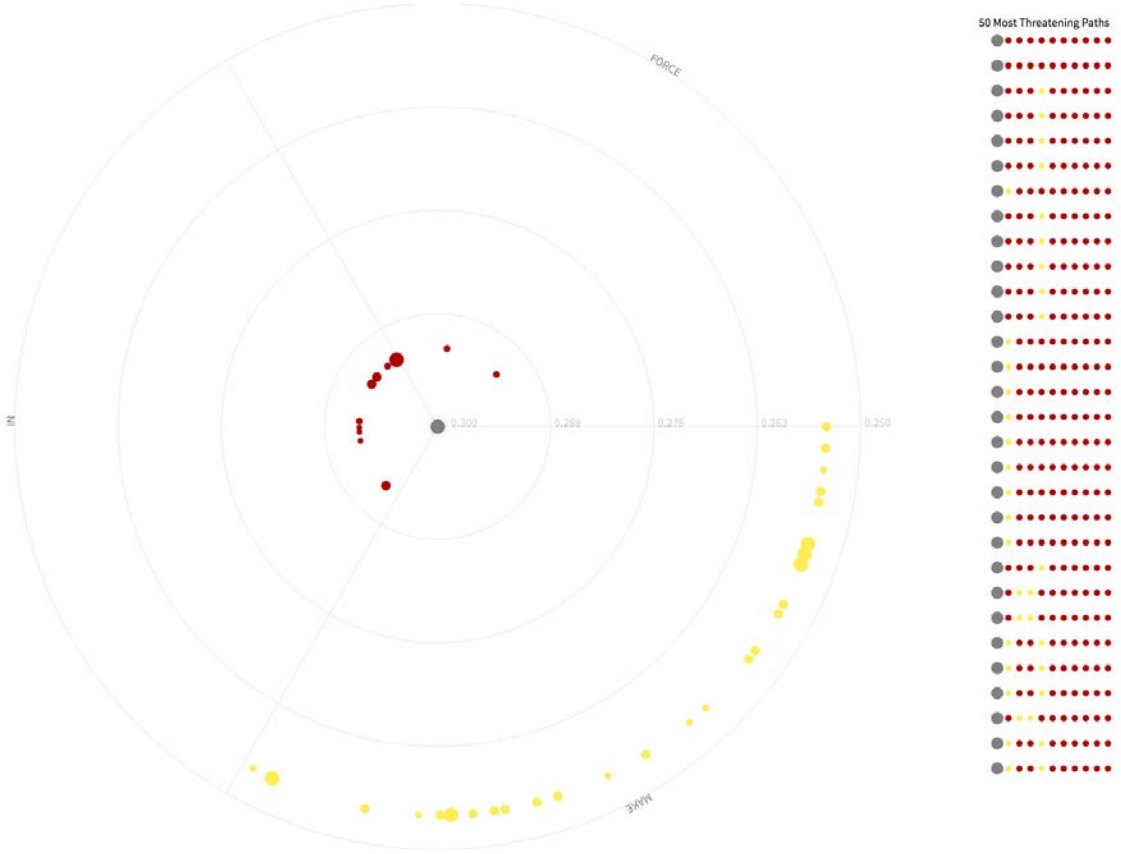


Figure 4.27.: Visualisation based on the Cloud case study that was first modelled in the ANM. The ANM generated an attack tree via the TreeMaker tool, and converted into linear attack paths. In this example there are three main actions on which the nodes are ordered. The scale of the visualisation automatically adapts to the data presented, here the scale is between ,30 and ,25. Also, most values in this example were default values, resulting in only two colours (yellow and red), while all shades between those colours would be possible. Interactive version at `lustlab.net/dev/trespass/visualizations/at-sketches/cloud_cluster.html`.

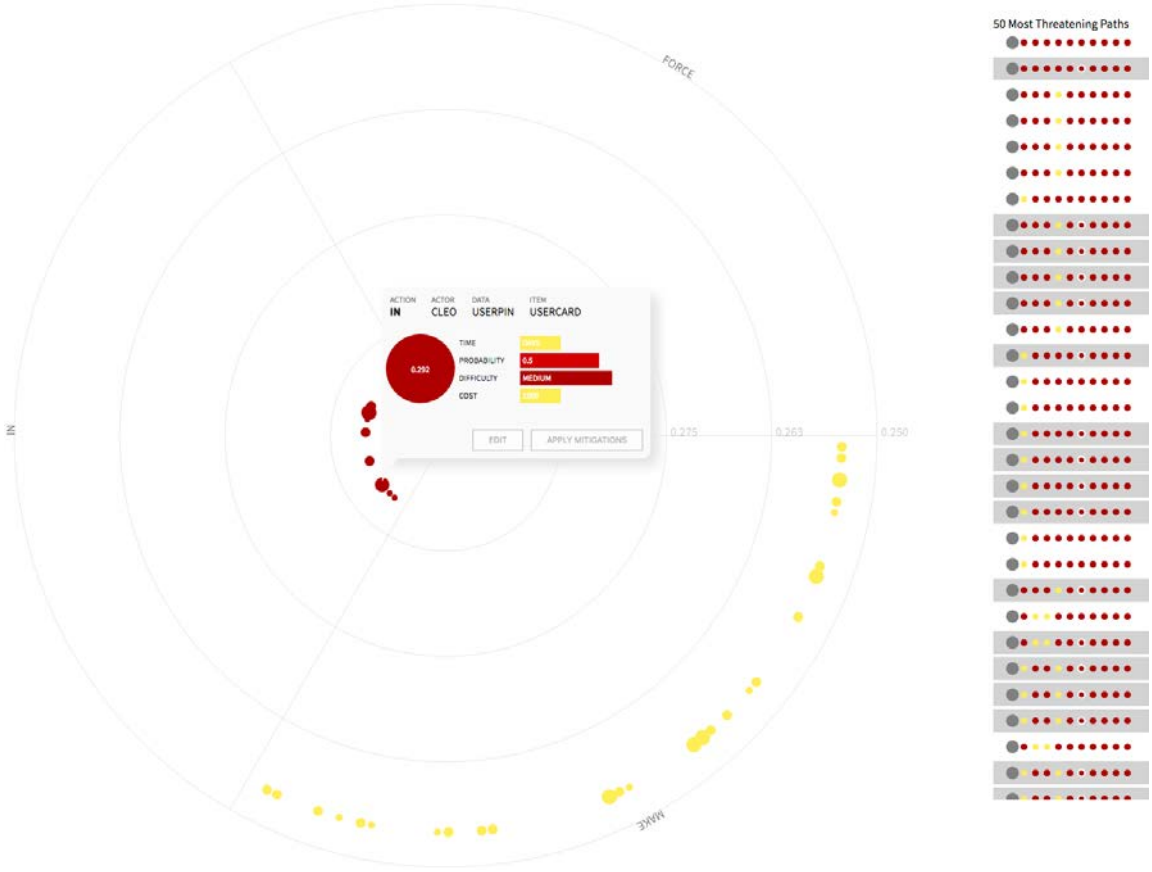


Figure 4.28.: In this figure, a user hovers over one of the nodes and a large tool tip label appears. This tool tip contains the label describing the action attached to the node, in the form of a circle that indicates in colour and number the general threat score of this node, and a bar graph with the four parameters the general threat score is composed of. Clicking on the node makes the tool tip editable, and the user can edit or apply mitigations to the node. These are applied by changing the parameters in the tool tip or by dragging the node to a different position in the graph, updating the parameters. On the right, the 50 most threatening paths are displayed. On hover over the node in the main graph, those paths are highlighted where this node appears. To highlight the exact position of the node in the attack path, the node in question is given a white circle.

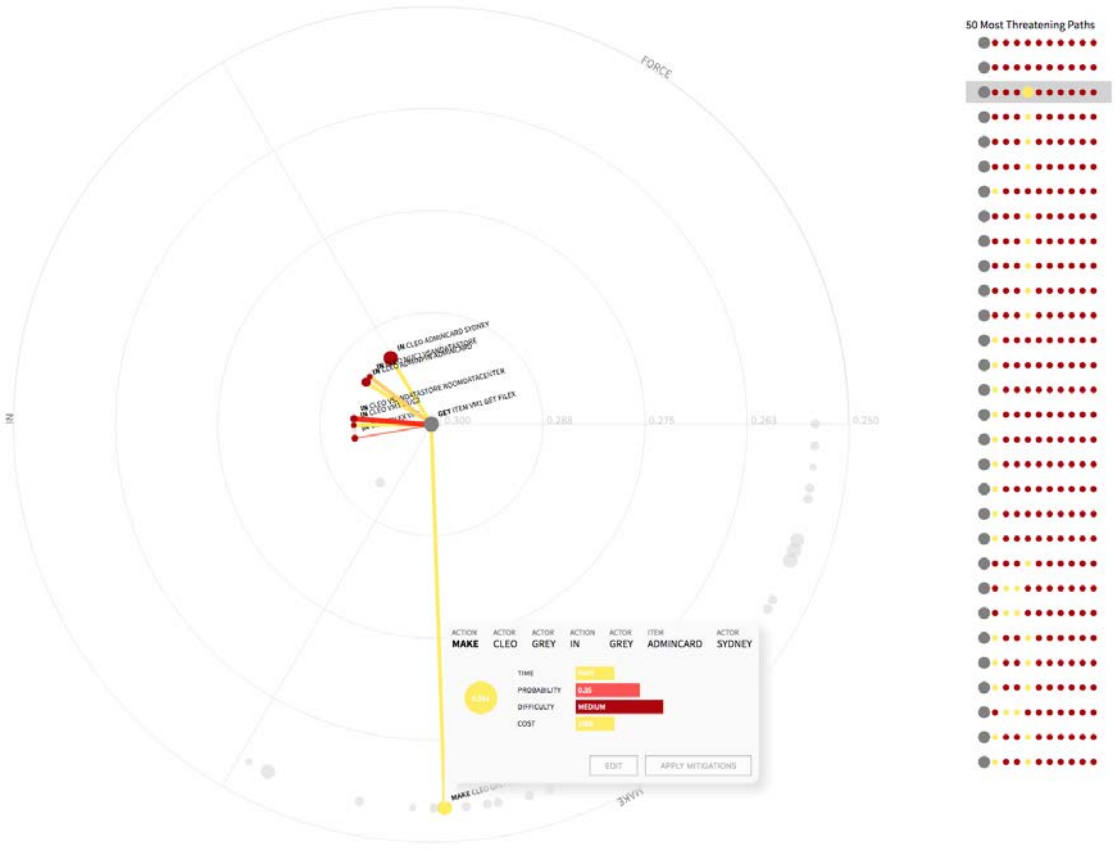


Figure 4.29.: In this figure, a user hovers over one of the nodes in the column on the right side of the graph. This "attack path" is highlighted with a grey bar, and the corresponding node appears with a white outline. All nodes that are part of this path draw lines to the ‘goal’ node in the middle of the graph. The colour of these paths indicate difficulty, the colour indicates time, as can also be seen in Figure 4.26.

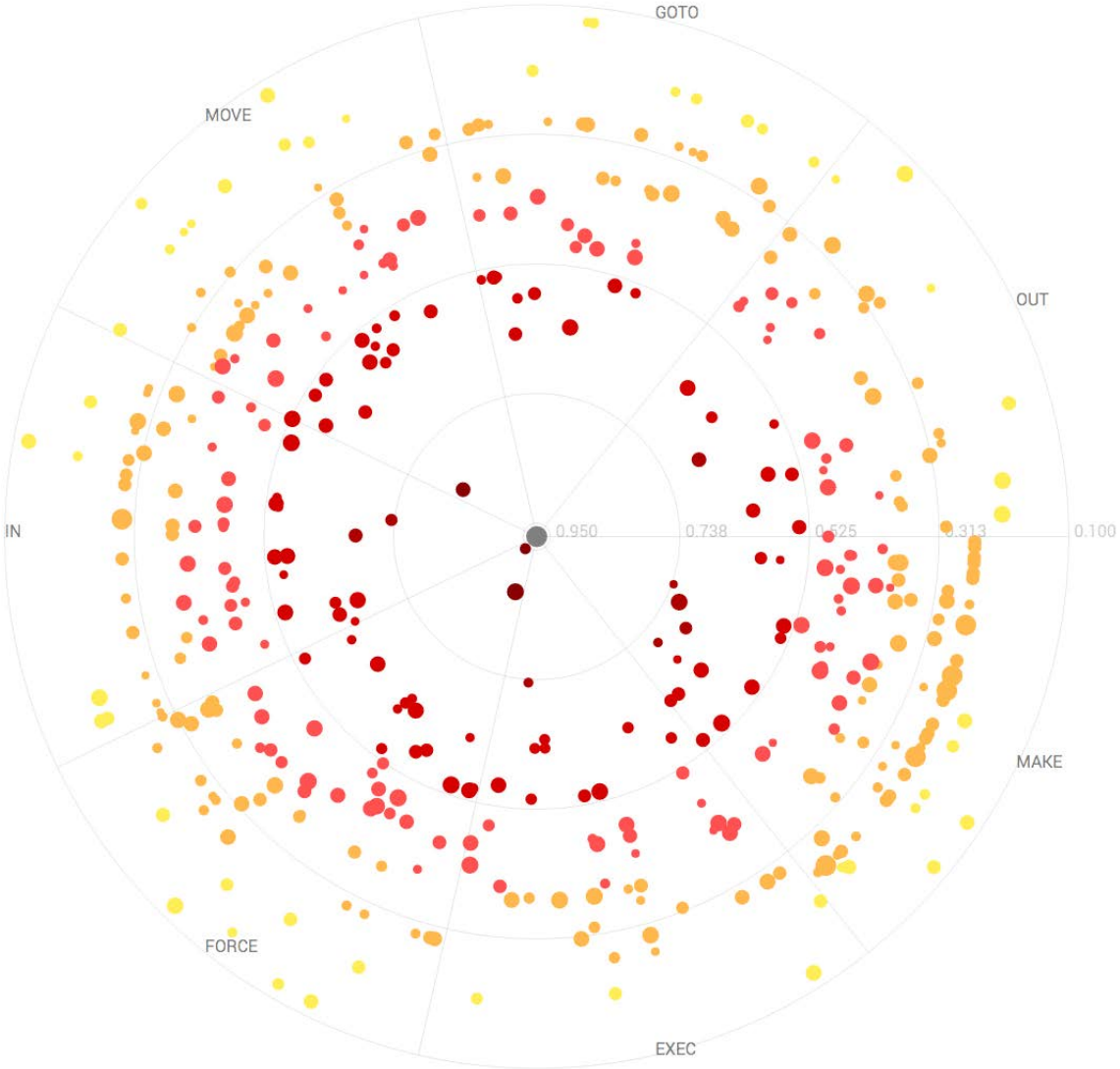


Figure 4.30.: Example of an attack cloud based on a large attack tree with as main actions "go to", "out", "make", "execute", "force", "in", and "move".

4.6. Visualisations for the ATM case study

The ATM (Automated Teller Machine) case study aims to study attacks to these kinds of machines, which can be from:

- Software attacks consisting of infecting the machines with malware software that allows the attacker to take control of the devices, including the ability to open the machine money vault, and record data from the cardholders;
- Physical attacks consisting in stealing the machines to open them in order to access the money vault.

Securing automated teller machines (ATMs), as critical and complex infrastructure, requires a precise understanding of the associated threats. The ATM case study tries to capture the most dangerous multi-stage attack scenarios applicable to ATM structures. This is done through creating attack-defence trees to model and analyse the security of ATMs. Based on expert knowledge and available historical data, the attack-defence tree have been decorated with estimations for critical parameters, such as likelihood of an attack. Next to this, the ATM case study partners have also collected a large data set on ATMs in Lisbon, including data on attacks over the last five years, locations, loss, victims, type of attacks, data on locations of entrances to highways, locations of police stations, unemployment rates in neighbourhoods, and many more.

Visualising geographical information The visualisations of geographical information of ATMs in Lisbon have the aim to show possible correlations between geographic locations and detailed data about the identification of the attack. For instance, the position of an ATM in combination with the distance to the entrance of a highway, and the distance to a police station are visualised as having an influence on the vulnerability of an ATM. This is combined with data about attacks on the ATMs themselves.

Each map is composed of layered information:

- Geographic locations of ATMs
- Area of vulnerability
- Frequency of attacks

A second layer alternates different data:

- Geographic locations of police stations
- Geographic locations of highways and main routes
- Number of successful attacks
- Distinction between manual attacks and logical attacks
- Attack duration
- Distinction between gross loss and indirect loss

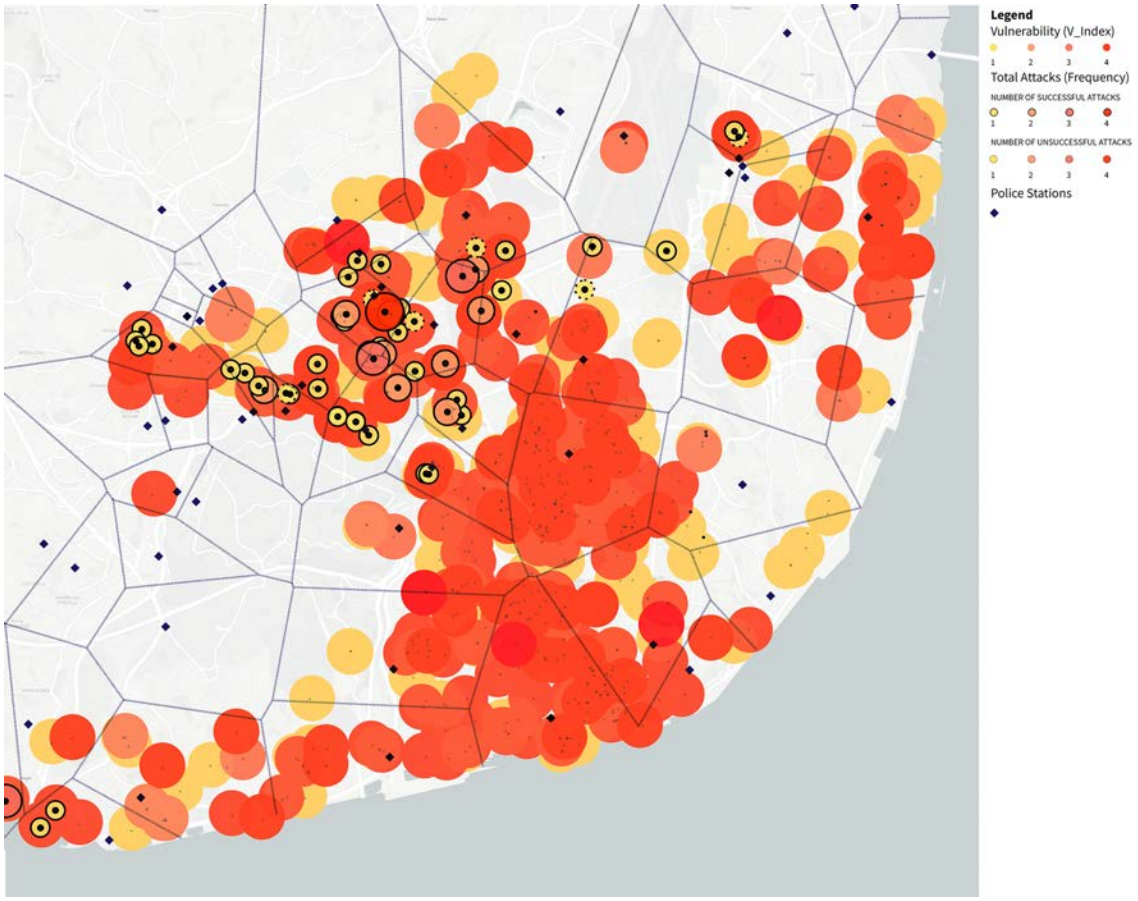


Figure 4.31.: Visualisation based on 724 ATM points in Lisbon and their distance to the entrance of the highway versus distance to police stations. On top of this data a set of 121 attacks on ATMs is plotted.

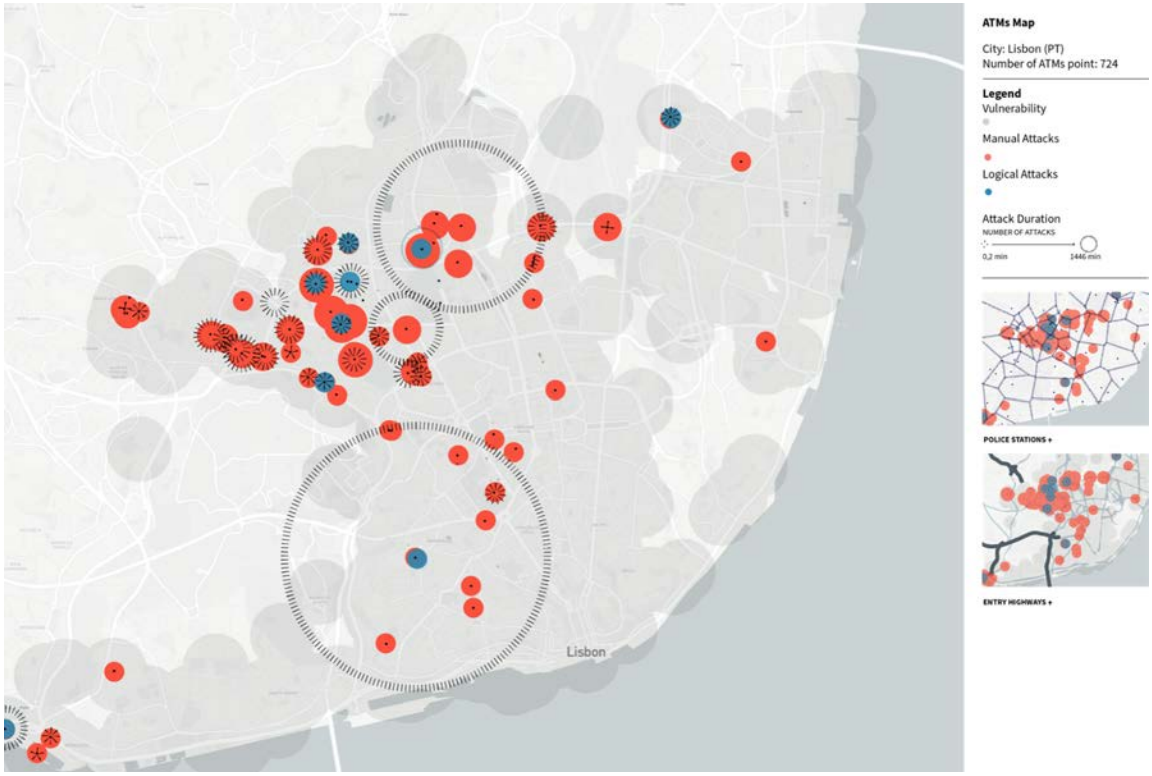


Figure 4.32.: Visualisation of attacks, split into manual attacks and logical attacks, plus the time the attack took. The small sub maps on the right show context with distance to highways and distance to police stations. See all ATM visualisations at visualisation.trespass-project.eu/?tag=geo.

4.7. Application to attack graphs

Attack graphs are a common tool used by security researchers to organise information on all possible attack paths within a certain space. Although they generally are adapted for custom use, the general idea is the same: there is a directed graph with a starting point and an end point (the goal), as well as nodes that function as attack steps or entities (Fig. 4.33) (as used in (Bassett, Gabe and Verizon Enterprise Solutions, 2015)). The edges are possible paths from one entity to another. These nodes and edges carry with them several parameters, such as probability, cost, and incident count.

For the remainder of this section, however, the attack graph referred to is the one defined by Verizon in their annual Data Breach Investigations Report (DBIR)[7] (Verizon Enterprise Solutions, n.d.) as a test case to see whether we could also apply the visualisation principles laid forward in the preceeding chapters to graphs other than attack trees. For 2016, there are seven action groups, with multiple sub-actions, as well as three attribute groups, again with multiple sub-attributes. Within the graph itself, actions lead to other actions or to compromised attributes. Compromised attributes will lead either to the end of the breach or to another action by the attacker.

Visualising attack graphs There are several goals that the visualisation of the attack graph aims to achieve: (i) displaying and differentiating actions and attributes, (ii) displaying relative threat of nodes and edges, (iii) displaying paths, and (iv) displaying a comparison between different versions of the graph (either through mitigations or comparison with previous years' data). The principal flaw of traditional attack graph visualisations is that they attempt to visualise all nodes and connections at once. In cases such as the DBIR, this grows very complex and as a result, it becomes hard to perform even simple tasks, such as determining the relative importance of a node or discovering which nodes are connected. In fact, the version presented at (Bassett, Gabe and Verizon Enterprise Solutions, 2015) mostly serves to illustrate how complex the attack space is.

As the principal structure is a directed graph, it is possible to immediately identify the nodes and edges as elements of its vocabulary. Digging deeper, the following characteristics make up these elements:

Node	**Edge**
Type of node (action, attribute, start, end)	Type of edge (edge to attribute, action, or end)
Relative frequency of node (occurrences within incidents)	Relative frequency of edge (occurrences within incidents)
Sub-type of node (one of 5 actions or one of 3 attributes)	

[7] Data link: https://github.com/vz-risk/VERISAG/tree/v2/static.

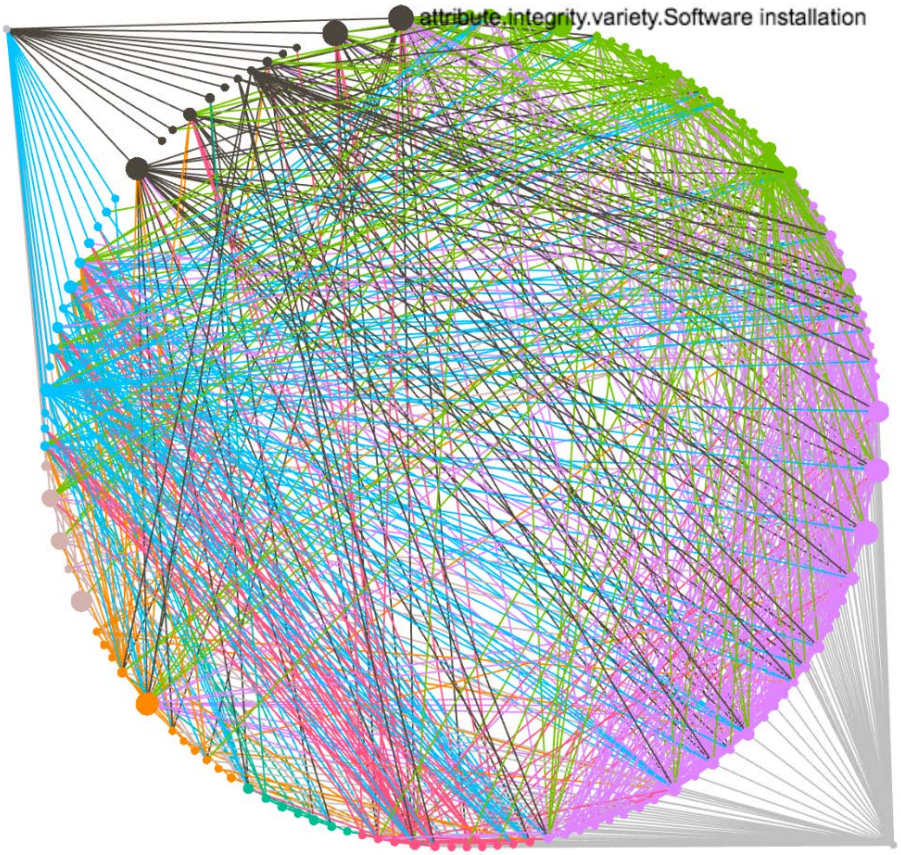


Figure 4.33.: Example attack graph as used in the online tool 2016 DBIR Attack Surface Analysis.

The visual language begins with the same traditional elements of the directed graph: nodes and directed edges. Traditionally, these graphs are visually composed of circles that represent the nodes, and paths with arrows indicating direction. To begin building a visual vocabulary, each of the elements is parameterised. Assigning radius and fill colour of the circle to represent frequency of incidents creates an aesthetically informative visualisation of the node. Textual treatment and visual treatment can then be applied to each circle to indicate the type of node. Rather than by drawing an arrow as a path, direction on edges can be shown by decreasing the stroke width of a path. This width can also be parameterised, as well as the opacity of the edge, to show how frequently that edge occurred within the incident space, resulting in the legend in Fig. 4.34. From this legend the visualisation[8] was developed based on the outlined approach.

[8]The interactive version of the DBIR Attack Graph can be found at http://lustlab.net/dev/vzw/index.html.

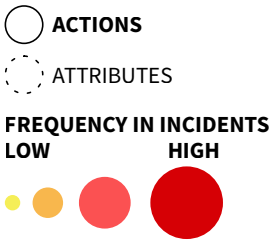


Figure 4.34.: Legend for attack graph visualisation.

Arc diagram Arc diagrams were chosen because of their ability to clearly display multivariate data, even in complex situations. Building on the work of (Wattenberg, 2002), these graphs provide an easy way to visualise connections by placing the nodes on the same line (Fig. 4.35). They also allow easy comparison of nodes and edges, and give viewers a more linear ability to visualise this data. The two-sided nature of this graph allows one to visualise edges going to attributes on one side, and edges going to actions on the other, providing a clear visual delineation between the two types of edges. When a comparison of different versions of the graph should be shown, all edges can be moved to one side and two versions of the graph can be shown simultaneously.

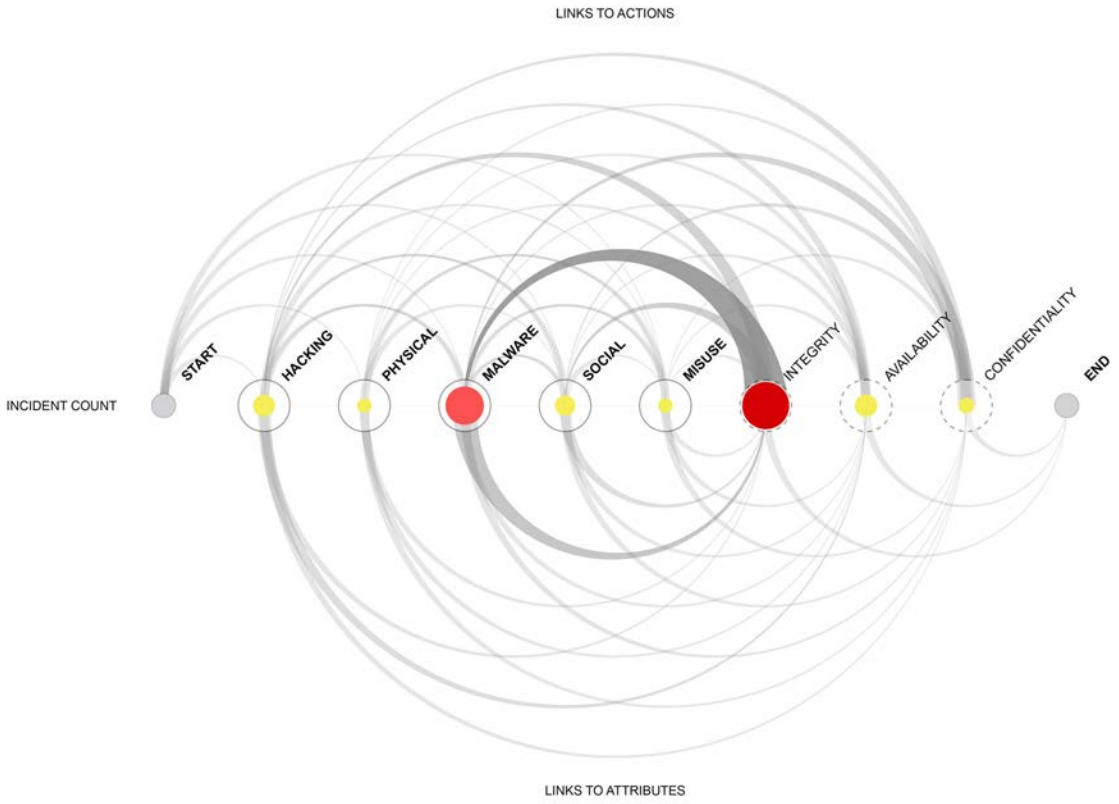


Figure 4.35.: Macro-view of 2016 DBIR Data.

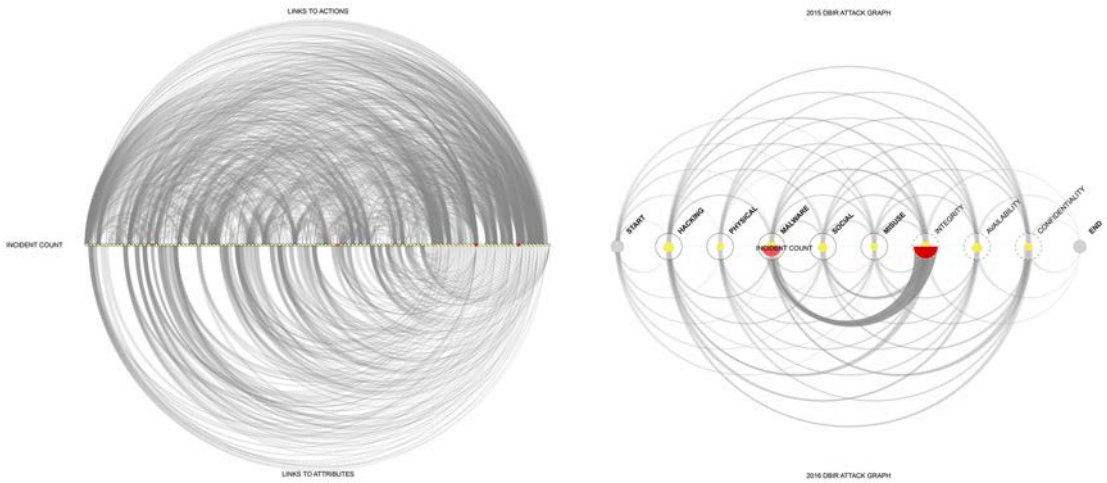


Figure 4.36.: More visualisation views afforded by using an arc diagram.
Left: All nodes of the 2016 DBIR.
Right: Comparison between 2015 and 2016 DBIR.

4.7.1. Semantic zooming

Although visualisation of all 119 nodes in one view provides a good overview, this level of complexity is very hard to follow and becomes unactionable. To improve this, semantic zooming can be applied. The initial view now becomes the eight macro categories of actions and attributes. This presents a good overall sense of how certain attacks paths might be structured, as well as the relative frequency of certain action or attribute categories within the attack space. If viewers want to learn more about the composition and frequency of each attack or attribute, they can click on it for a view containing all subgroups (Fig. 4.37a). We choose to contain these sub-nodes within the overall node to visually show the hierarchy of nodes. At this level, viewers can see the count of each node and determine which specific sub-node presents a greater threat. This also applies to comparing graphs (Fig. 4.37b).

4.7.2. Contextual awareness and highlighting

A level of interactivity is also built into the visualisation that allows the viewer to highlight certain aspects of the graph depending on the current zoom level. At the macro-view, hovering over a node reveals the incident count of that node within the attack space, and other child nodes (Fig. 4.37c). This allows viewers to pay attention to the context in which they are highlighting the node. All other nodes are greyed out for clarity, which further aids focusing on the relevant information. In the micro-view, when a user hovers over a node in the graph, the incident count at the bottom of the circle updates to also display information specific to the currently highlighted node (Fig. 4.37a and b). Visual feedback is also given by turning the highlighted node white to match the text colour. This contextual

highlighting provides information to the user only when requested and presents the wealth of information in the graph in a non-overwhelming manner.

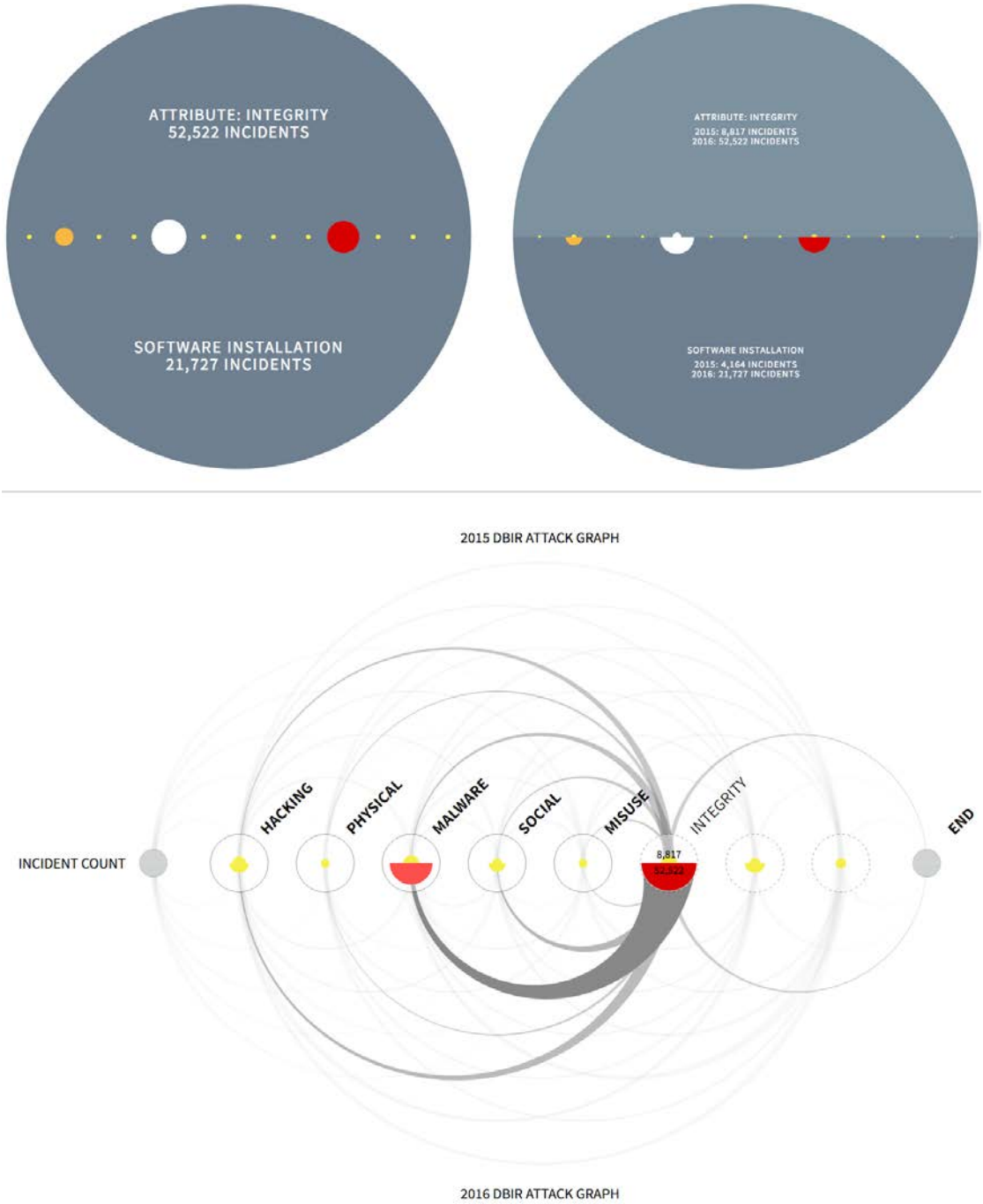


Figure 4.37.: From left to right: (a) Micro-view of 2016 DBIR data with highlighting, (b) Micro-view of 2015/2016 DBIR data with highlighting,
Under: (c) Highlighting a specific node and its target edges in the 2016 DBIR.

5. Engagement, impact and exploitation

In this chapter we present the main engagement activities conducted by WP4 and summarise the short-term impacts as well as the potential for longer term impacts and exploitation.

5.1. Engagement activities

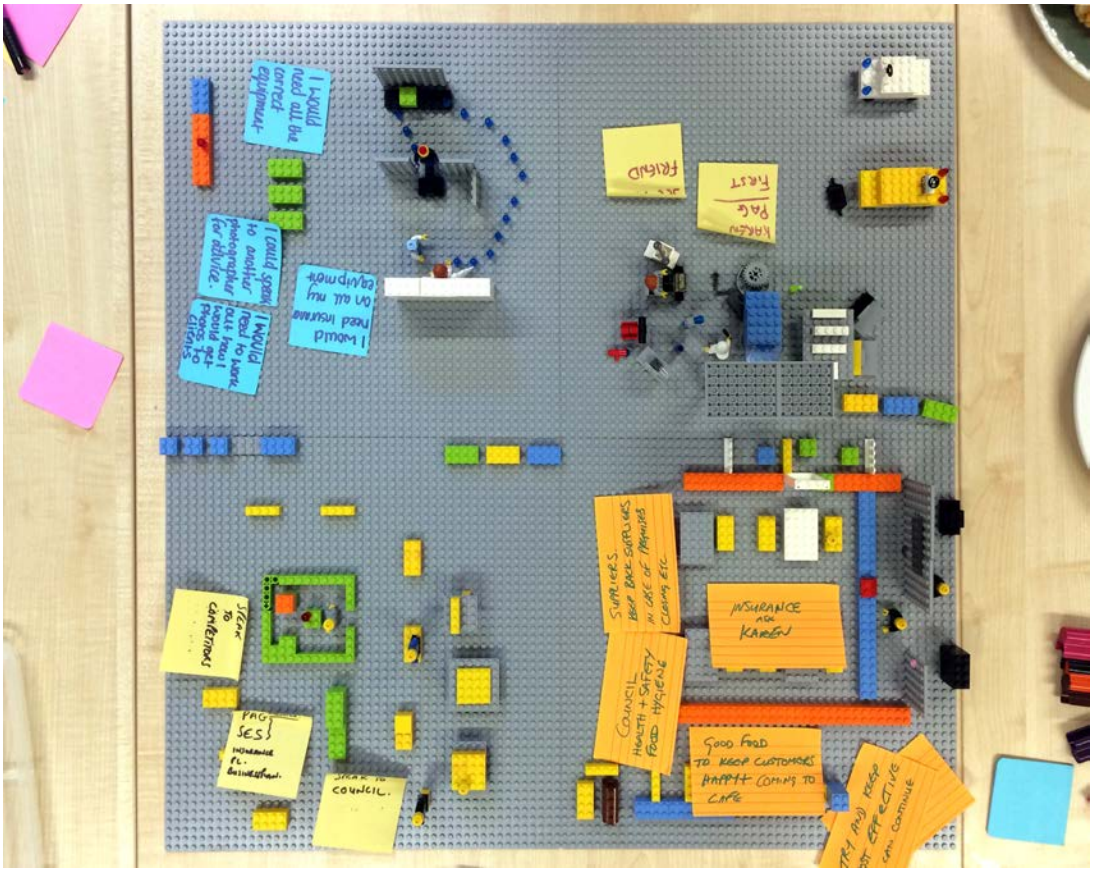


Figure 5.1.: Sunderland workshop 2105: the four models (made by five participants) joined together during the later part of the session and joint discussions, and seen from above. Clockwise from top left, the businesses ideas represented are photographer, car restorer, restaurant, and taxi/delivery/removals driving business. The notes seen were added during post-modelling discussions.

5.1.1. Visualisation workshop for SMEs 2015

A workshop was carried out at the Pallion Action Group (PAG) in Sunderland in June 2015. PAG is a proactive community centre, which acts as a valuable source of information for community members and is able to connect community members to other providers of advice, resources, and assistance where this is needed. It is a trusted gatekeeper both to and from the local community. The workshop we provided was designed for newcomers to business to bring their ideas for start-ups, in order to more carefully think through their proposed businesses in a safe and supportive environment, such that their early-stage ventures do not stumble upon key aspects of data management and cybersecurity.

The small groups and individuals and small groups were given access to LEGO kits and guided through the process of constructing their business model in tangible form. Among the many points that were raised by the participants„ a number of common threads emerged, to be fed back to PAG and other interested stakeholders:

The importance of a mentor for an individual or small group to be able to refer essential questions to. This should cover important issues related to provision of insurance, satisfying statutory requirements regarding public liability, the management of premises, as well as the heath and safety implications of business ideas. Other key issues noted were those that might arise due to implementation of apprenticeships, internships, and the design and application of business plans. Mentors, it was envisaged, should be able to address these questions in a timely way, either by providing information directly, or pointing to other partners. It was thought that it would be preferable to have a singe dedicated mentor for each new venture, and this could be coordinated by PAG.

Single points of data and information-access failure were identified in some of the envisaged arrangements described by the group. These were easily remedied by simple countermeasure steps. For example, ensuring that a customer database is backed-up to a secure location, one that can be accessed in the event of the loss of a mobile handset, for example. In other ways, a ‘back-up plan’ would also be needed, for example in supply and delivery chains, and short-term staff booking systems. There was also an acknowledgement that their initial business concept, and the business plans that accompany them, would need to be adaptable, and that in the process of finding an appropriate and sustainable niche for their business would be a progressive matter, developed through trial and discussions with potential customers in the early stages.

This coordination of mentorship roles, was an overview of a present situation where there are ever-increasing requests directed towards the Pallion Action Group for business start-up information and support. PAG’s resources are inevitably limited, and the meeting acknowledged that dedicated attention from a mentor would alleviate this situation and prevent any potential oversights and omissions in the provision of this information, and in the subsequent follow-up processes. This was also observed to be an opportunity for the SMEs to support each other through the mutual offering of services and to provide information security advice in context. In all this, PAG’s goal is to ease the difficult transition into self-sustaining trading, thus supporting people as they become less dependent on

state-provided benefits. It was clearly identified during this workshop that information security issues are present during the start-up stage and that these risks must be attended to for the ventures to the self-sustaining.

The workshop was deemed a success in promoting this goal, and the resulting models made by participants proved to be a rich source of discussion and in particular as a way of examining different perspectives on the scenarios they were modelling (Fig. 5.1).It was also demonstrated that LEGO modeling was an effective means of identifying potential information security risks to a non-technical audience.

A further SME workshop was run in Sunderland in October 2015, this time at the Sustainable Enterprise Strategies centre. This workshop looked at the provision of housing services and used LEGO as a visualisation method to identify information security risks to the stakeholders involved in the delivery of housing services. Similar results to the first SME workshop were delivered and physical modelling was demonstrated to be an effective visualisation method in this context.

5.1.2. Visualisation competition 2015

From June-October we organised a competition for visualisations that capture the social and technical complexity of so-called "cyber attacks". A special micro-site, social media, email and face-to-face engagement were used to promote the competition. The link was tweeted 109 times for the visualisation micro site, reaching 130.000 people in terms of impressions. Targeted emails were sent to four mailing lists (two computer security related lists and two design lists). In addition, individual emails were sent to three interaction design departments and two information security groups in the UK. Two interaction design departments were contacted in Australia. The call for participation was also circulated to the circa 3,000 alumni of Royal Holloway, University of London's Information Security Group.

All visual academies in the Netherlands were contacted, four presentations were given to design students to engage them: Two presentations were given at Artez Arnhem, interaction design department, Artez Enschede, cross media department, CMD Hogeschool Rotterdam, interaction design department, KABK The Hague, graphic design department.

In addition five visualisation specialists of high international standing were asked if they would both judge and promote the competition. All accepted and the judging panel consisted of Claude Heath, Rafael Marty, Lorraine Gamman, Manual Lima and Ben Fry.

Entries of Bente Brunia: *Hacking has never been easier, select your personal virus now*, and AlexOnline's: *Kinetic Visualization of the Social and Technical Complexity of Cyber Attacks*, were ranked equally, and won second prize.

Makayla Lewis won first prize with her entry: *Cyberstalking: its about control, not only privacy!* Jury member Raffael Marti on Makayla Lewis' poster: "Social issues around privacy and security are under appreciated and are not discussed enough. Only once it is too late do we realise how vulnerable we left ourselves to all kinds of unwanted exposures. Makayla's visual story does a good job helping create awareness around the problem."

Also for the two second place winners the jury members saw real-life applications, for instance to raise awareness for cyber attacks. Mr. Marti stressed that "engaging visualisations of modern cyber attacks helps communicate and understand just how intertwined todays systems are and how large the attack surface really is."

The winning entries all showed the importance of strong narrative in a visualisation and the need for cyber security to develop compelling and relatable narratives with which to communicate and engage on topics of risk.

Conclusions from the visualisation competition The results of the visualisation competition show the importance of narrative to successful visualisations. The judges represented a broad spectrum of both visualisation background and focus.The judges also represented different disciplines. It is noticeable that regardless of background or discipline, the judges all scored highest the visualisations with the strongest narratives.

The winning visualisations were strong partly because of the richness of the narrative that each visualisation communicated and partly because of the design of the visualisation for a particular audience.

These observations relate to the general visualisation principles that this work package defined in its first deliverable. These observations also reflect the importance of these principles for the development of the Attack Navigator Map.

As can be seen from the description of the publicity programme for this visualisation competition, much work was undertaken to engage with both the design and information security communities. Prior to the publicity programme, the call for participation took a long time to develop because it was developed for two communities. Despite this effort, it was still difficult to reach the relevant communities. This indicates that perhaps we should have run two separate tracks to the competition. It also reflects the difficulties of framing information security risk in such a way that it is accessible to both computer scientists and designers. See Appendix C for complete documentation.

5.1.3. Advanced visualisation workshop

On 22 and 23 September TRESPASS organised a Data Visualisation workshop. The first day was organised in liaison with WTHX[1], a one-day mini-festival around the topics Peace, Justice, Security + Code. TREsPASS adopted the security track and Paolo Ciuccarelli (Scientific Director Density Design and initiator of RAW[2] gave a keynote lecture and during that day participants worked in short sessions to formulate questions and prototype potential solutions for tomorrow's (security) problems.

During WTHX, professionals from the fields of peace, justice and security met coders, developers, artists, philosophers and designers. WTHX ('WhatTheHacks') is a crossover between a hackathon and a think tank. In small multidisciplinary teams more than 100

[1] http://www.wthx.org
[2] raw.densitydesign.org/

thinkers and makers assembled for twelve hours of ideation and co-creation during which they formulated questions and prototyped potential solutions for tomorrow's problems. An integrated programme led by experts provided depth to broaden ideas and views and stimulated the participants to come up with fresh answers. WTHX stimulates curiosity, expands new imagery and triggers participants into realizing new solutions and collaborations.

The second day was entirely dedicated to a data visualisation workshop, focusing on how to make visualisations around security that create impact, lead by Paolo Ciuccarelli, Michele Mauri, and LUST. The 35 participants were security practitioners, data visualisation specialists, students interaction and graphic design, journalists, etc..

The participants worked with data from the ATM case study, the geographical data set on ATMs in Lisbon, their attacks and all kinds of social data around this. In the introduction to the workshop, the goals were made clear and many tools for quickly getting visual insight in data sets were introduced. The participants worked in groups of 5-6 and during the day presented updates and ideas on the narrative that they found in the data set.

Conclusions from the advanced visualisation workshop The different groups all had very different perspectives on the data set, and many interesting narratives were created. One interesting example was the extraction of the number of victims per ATM that was attacked, in one case more than 145 victims that got skimmed. Their narrative tried to make the abstract data personal again by relating it to humans again. Also the difference between visualisations that were geographically-based and others that were more graph-based made very clear that data that has geographical qualities does not always need to be visualised geographically to get most impact.

It was very clear that from the participants, the security practitioners had the most difficulty in visualising the data. As they were not used to thinking in visual strategies they stayed closer to the data and could not easily create a narrative out of it that could lead to truly new insights. The cooperation with other professional fields helped getting them out of their comfort zone. Please see Appendix D for images.

5.1.4. TRE$_S$PASS Summer School visualisation workshops

The TRE$_S$PASS 2016 *Summer School on Social Aspects of Cyber Security Risk* sought to explore these challenges through a combination of high profile talks on the social aspects of cyber risks and hands-on workshops to transfer a range of modelling and analytical skills innovated specifically for the cyber security terrain. The speakers came from a range of academic disciplines including law, geography, sociology, politics and international relations, information systems and information security.

Two programmes of visualisation workshops were run on each afternoon of the Summer School. The purpose of these visualisation workshops was to help students synthesise the knowledge imparted through the talks and construct their individual knowledge base of the social aspects of cyber security risk.

Figure 5.2.: Summer School, RHUL, 2016. A group models a 'smart home' scenario with LEGO, while beside them an analyst transfers the actors, assets, and attacker goals from their physical model into the ANM on a laptop.

Having established in the practitioner panel and paper prototyping evaluation sessions that providing a risk context was important, the topic of cyber security risk and smart homes was selected. Smart homes are often looked at as the wave of the future, but the proliferation of the technology should always come with a word of caution. People who adopt smart home technology need to know about some of the biggest security concerns revolving around not only their appliances but also the central hub for controlling these gadgets e.g. smart phone. Workshop participants were asked to use a participatory visual research method (PVRM) to explore the security risks of a smart home device. The following visual techniques were offered:

- Physical modelling using TRE$_S$PASS LEGO techniques.
- Storyboarding using a cartoon-making kit.

Using the TRE$_S$PASS LEGO modelling techniques, we were able to demonstrate that a brainstorming session can be conducted in tandem with a TRE$_S$PASS consultant building a model of the LEGO scenario in the ANM. This finding parallels our finding from the UK postgraduate evaluation session where students brainstormed an access control scenario and used the paper prototyping kit to make a map of the LEGO scenario and conduct attack tree analysis on that map. Thus the steps the groups were able take ran from physical modelling and discussions (Fig. 5.2), then inputting actors and other features of their models into a new ANM model of the scenario (Fig. 5.3), and finally running an analysis on this model and evaluating the resulting attack trees (Fig. 5.3).

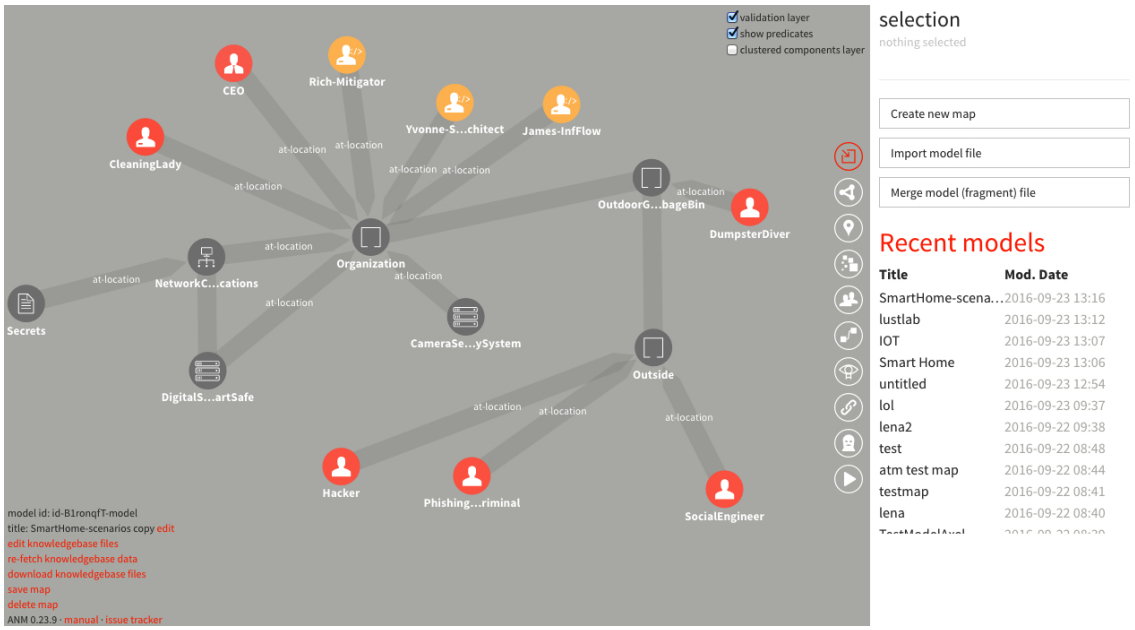


Figure 5.3.: Summer School, RHUL, 2016. The ‘smart home’ scenario having been modelled with LEGO, is entered into the ANM as a new model and scenario.

Figure 5.4.: Summer School, RHUL, 2016. The ‘smart home’ scenario having been modelled with LEGO, produced an initial attack tree.

5.2. Impact

We have conducted a number of key outreach activities where we have deployed our visualisation principles and techniques. Perhaps the most notable is our involvement with the Verizon Data Breach Investigations Report. Our work here is best summarised with this quote from Verizon:

> "We recently worked with you to visualise an attack graph for the 2016 Verizon Data Breach Investigations Report. For the ninth time, the Data Breach Investigations Report (DBIR) lifts the lid on what's really happening in information security. The 2016 dataset is bigger than ever, including over 100,000 incidents and examining 2,260 confirmed data breaches across 82 countries. With data provided by 67 contributors including security service providers, law enforcement and government agencies, this year's report offers unparalleled insight into the information security threats."

Verizon wrote to us and described the take-up of our visualisations in the following way:

> "To date we can see that tens of thousands of readers have engaged with your contributions. The report is quoted extensively by top tier and business press throughout the year as well as frequently cited in government, industry, and academic speeches. Your contributions have enabled us to communicate the complex concept of networked risks in ways that were previously difficult using our traditional approaches."

In their feedback Verizon also highlights the potential for longer term impact:

> "Going forward we would like to use the arc graph visualization approach to continue to help change the way information security professionals think about risk. Attackers already think in terms of graphs; however defenders tend to think in terms of lists. With your help, we look to change that."

We have also achieved impact with our use of physical modelling tools during the life of the project. This is documented in (The TRESPASS Project, D4.3.3, 2016)

5.3. Exploitation potential

From our engagement and impact work, we have developed an exploitation plan that reflects a number of potential outlets for our work.

Inclusion as part of UK government's socio-technical modeling initiatives and engagement guidance We have engaged with The UK government's National Technical Authority for Information Assurance (CESG) during the lifetime of this project and have demonstrated both the ANM, InterActor and the physical modelling process and techniques. Discussions are on-going as to how TRE$_S$PASS elements support the UK's strategy for socio-technical security modeling and security engagement.

Security awareness and engagement platform We have engaged with a number of RHUL's alumni about the possibility of adopting TRE$_S$PASS visualisation principles, Atlas and icons as part of new approaches to security awareness and engagement platform. These discussions are on-going and in two cases are at the prototype evaluation stage.

Security reports and communication Following on from our success with the Verizon Data Breach Investigations report, we continue to develop our journalistic visualisation capability and have interest from Verizon to continue the partnership in next year's data breach report with a view to making the attack graph format the de-facto visual form for such reports.

6. Conclusions

During the TRE$_S$PASS project we have developed a number of significant contributions to the state of the art in risk visualisations. The main contributions include:

- A visual editor, based on the state of the art open-source graphical representations, that is specifically designed to visualise attack tree analysis;
- A radial visualisation that represents attack tree analysis and enables drill-down and highlighting to show particular aspects of analysis;
- A set of risk visualisation principles and rules that have been evaluated by more than 100 participants over a four year period;
- An alternative visual form to the tree structure, specifically designed to show attack paths in the cloud environment; and
- Visual principles that can articulate risk elements from the social or physical realm combined with risk elements from the digital realm.

In addition, we have developed a paper prototyping tool kit specifically for the purpose of helping organisations to better understand their risk scenarios. We have deployed this tool kit both in teaching, in risk assessment and in product design.

We have further extended the state of the art in visualisations of social aspects of risk (The TRE$_S$PASS Project, D4.3.3, 2016) and in responding to specific aspects of complexity over time (The TRE$_S$PASS Project, D4.3.2, 2016). These contributions are documented in their relevant deliverables.

The exploitation potential for these contributions has been documented in the previous chapter and reflect the durability as well as the novelty of what we have achieved.

References

Akama, Y., & Ivanka, T. (2010). What community?: facilitating awareness of 'community' through playful triggers. In *Proceedings of the 11th biennial participatory design conference* (pp. 11–20).

Alberts, C. J., & Dorofee, A. (2002). *Managing information security risks: The octave approach*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Barber, B., & Davey, J. (1992). The use of the CCTA Risk Analysis and Management Methodology (CRAMM) in Health Information Systems. *Medinfo*, *92*, 1589–1593.

Bassett, Gabe and Verizon Enterprise Solutions. (2015, June). *DBIR Attack Graph Analysis.* Retrieved from http://dbir-attack-graph.infos.ec/

Bederson, B., B. & Shneiderman. (2003). *The craft of information visualization*. Elsevier.

Bertin, J. (1967). *Sémiologie graphique*. Gauthier-Villars, Paris, France.

Boling, E., & Frick, T. W. (1997). Holistic rapid prototyping for web design: Early usability testing is essential. *Web-based instruction*, 319–328.

Chen, B., Avrunin, G. S., Clarke, L. A., & Osterweil, L. J. (2006). Automatic fault tree derivation from little-jil process definitions. In *Proceedings of the 2006 international conference on software process simulation and modeling* (pp. 150–158). Berlin, Heidelberg: Springer-Verlag.

Cockburn, A., Karlson, A., & Bederson, B. B. (2008). A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)*, *41*(1), 2.

Drucker, J. (2014). *Graphesis*. Harvard University Press.

Eppler, Martin J. and Aeschimann, Markus. (2008). *Envisioning risk: A systematic framework for risk visualization in risk management and communication.* Retrieved from http://www.knowledge-communication.org/pdf/envisioning-risk.pdf

Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, *35*(3), 379–383.

Flach, J. (2012, September). Complexity: learning to muddle through. *Cognition, Technology & Work*, *14*(3), 187-197. doi: 10.1007/s10111-011-0201-8

Frisch, M. (2012). *Visualization and interaction techniques for node-link diagram editing and exploration*. Verlag Dr. Hut.

Fry, B. (2007). *Visualizing data: Exploring and explaining data with the processing environment*. O'Reilly.

Harris, R. L. (1999). *Information graphics: A comprehensive illustrated reference*. New York, NY, USA: Oxford University Press, Inc.

Husdal, J. (2001). *Can it be really that dangerous? Issues in visualization of risk and vulnerability.* Working paper, University of Utah, Salt Lake City. Retrieved from http://www.husdal.com/2001/10/31/can-it-really-be-that-dangerous-issues-in-visualization-of-risk-and-vulnerability/

Jaspers, M. W., Steen, T., van Den Bos, C., & Geenen, M. (2004). The think aloud method: a guide to user interface design. *International journal of medical informatics*, *73*(11), 781–795.

Kalawsky, R. S. (2009). Gaining greater insight through interactive visualization: A human factors perspective. *Trends in Interactive Visualization*, 119–154.

Kipouros, O., T. & Isaksson. (2016). Visual analytics for evaluation of value impact in engineering design. *ECCOMAS Congress*.

Kirk, A. (2015, February). *References for visualising uncertainty.* Retrieved from http://www.visualisingdata.com/2015/02/references-visualising-uncertainty/

Koffka, K. (1922). Perception: An introduction to the gestalt-theorie. *Psychological Bulletin*, *19*(10), 531–585.

Koffka, K. (1935). *Principles of gestalt psychology*. Harcourt, Brace and Company, New York, NY, USA.

Liggesmeyer, P., & Rothfelder, M. (1998). Improving system reliability with automatic fault tree generation. In *Digest of papers: Ftcs-28, the twenty-eigth annual international symposium on fault-tolerant computing, munich, germany, june 23-25, 1998* (pp. 90–99). Retrieved from http://doi.ieeecomputersociety.org/10.1109/FTCS.1998.689458 doi: 10.1109/FTCS.1998.689458

Majdara, A., & Wakabayashi, T. (2009). Component-based modeling of systems for automated fault tree generation. *Reliability Engineering & System Safety*, *94*(6), 1076 - 1086. Retrieved from http://www.sciencedirect.com/science/article/pii/S0951832008002913 doi: http://dx.doi.org/10.1016/j.ress.2008.12.003

Marty, R. (2008). *Applied security visualization* (1st ed.). Addison-Wesley Professional.

Mazza, R. (2009). *Introduction to information visualization*. Springer-Verlag, London.

Norman, D. A., & Stappers, P. J. (2015). Designx: Complex sociotechnical systems. *She Ji: The Journal of Design, Economics, and Innovation*, *1*(2), 83 - 106. Retrieved from http://www.sciencedirect.com/science/article/pii/S240587261530037X doi: http://dx.doi.org/10.1016/j.sheji.2016.01.002

North, C., & Shneiderman, B. (2000, November). Snap-together visualization: Can users construct and operate coordinated visualizations? *Int. J. Hum.-Comput. Stud.*, *53*(5), 715–739. Retrieved from http://dx.doi.org/10.1006/ijhc.2000.0418 doi: 10.1006/ijhc.2000.0418

Parish, N. (2007). *Henri michaux: Experimentation with signs*. Rodopi, New York.

Park, K. E. . K. H., J. (2014). Design and implementation of the honeycomb structure visualization system for the effective security situational awareness of large-scale networks. *Journal of The Korea Institute of Information Security & Cryptology*, 1197–1213.

Raskin, A. (2010). *Privacy icons: Alpha release.* http://www.azarask.in/blog/post/privacy-icons. (Online; accessed 2014-09-30)

Rosenquist, Matt and Intel IT . (2009, december). *Prioritizing Information Security Risks with Threat Agent Risk Assessment.* Retrieved from http://www.intel.com/Assets/en_US/PDF/whitepaper/wp_IT_Security_RiskAssessment.pdf

Roth, F. (2012). Visualizing risk. Retrieved from http://e-collection.library.ethz.ch/view/eth:6286 doi: 10.3929/ethz-a-007580138

Schneier, B. (1999). Attack trees: Modeling security threats. *Dr. Dobb's Journal of Software Tools*, *24*(12), 21–29. Retrieved from https://www.schneier.com/

cryptography/archives/1999/12/attack_trees.html

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of the ieee symposium on visual languages* (p. 336 -343). doi: 10.1109/VL.1996.545307

Shneiderman, B., & Plaisant, C. (2003). *Designing the user interface*. Pearson Education India.

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

The TRESPASS Project, D4.1.2. (2015). *Final requirements for visualisation processes and tools.* (Deliverable D4.1.2)

The TRESPASS Project, D4.2.1. (2014). *Initial report on visualisations of information security risks.* (Deliverable D4.2.1)

The TRESPASS Project, D4.3.1. (2014). *Initial visualisations of socio-technical dimensions of information-security risks.* (Deliverable D4.3.1)

The TRESPASS Project, D4.3.2. (2016). *Visualisation to simplify complex information.* (Deliverable D4.3.2)

The TRESPASS Project, D4.3.3. (2016). *Visualisations of socio-technical dimensions of information-security risks.* (Deliverable D4.3.3)

The TRESPASS Project, D6.1.1. (2013). *Initial requirements for tool integration.* (Deliverable D6.1.1)

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

The TRESPASS Project, D6.3.1. (2015). *Prototype of the TRESPASS user interface.* (Deliverable D6.3.1)

Tidwell, J. (2005). *Designing interfaces: Patterns for effective interaction design*. O'Reilly Media.

Tufte, E. (1990). *Envisioning information*. Graphics Press.

Verizon Enterprise Solutions. (n.d.). *2016 Data Breach Investigations Report* (Tech. Rep.). Verizon.

Vigo, R., Nielson, F., & Nielson, H. R. (2014). Automated generation of attack trees. In *Computer security foundations.*

Ware, C. (2000). *Information visualization: Perception for design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Wattenberg, M. (2002). Arc diagrams: Visualizing structure in strings. In *Ieee symposium on information visualization, 2002.* (pp. 110–116).

Yi, J. S., ah Kang, Y., Stasko, J. T., & Jacko, J. A. (2007). Toward a deeper understanding of the role of interaction in information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, *13*(6), 1224–1231.

A. Report from feedback panels

Participants: Erik Otto (Shell-Senior Information Security Advisor), Rob Cordes (Freelance Security Officer), Arenas Mendoza Gloria, (KPMG-Advisor), Ivan de Wit, (KPMG-Junior Advisor), Hilbrand Kramer, (Nationale-Nederlanden-Senior ORM Officer), Ruud Verbij (KPMG), Jeroen Veldhoen (KMPG), Daan Spakenburg (Freelance Security Officer), Thris Williams, (Healthcare security specialist), Konrad Wrona (NCIA NATO-Principal Scientist), Daniel Datau (NCIA NATO), Peter Lenk (NCIA NATO), Philippe Lagadec (NCIA NATO), Reginald Sawilla, (NCIA NATO), Tamsin Moye (NCIA NATO), Michael Street (NCIA NATO), Marc Richter (NCIA NATO), Jeroen Barendse (LUST), Frederic Brodbeck (LUST), Robin Smits (LUST)

Attack trees

- Attack paths are accepted as being important, but it also necessary to show defences.
- Can certain properties be prioritised, and perhaps be weighted differently (with direct probable consequences for the visualisation process)?

Visualisation feedback The following points were identified as important to note for our future development:

- Countermeasures are missing, either existing ones or those planned.
  - Representing the impact of attacks should also be considered as a priority.
  - Also the cost of mitigation as compared to possible impacts.
- The ability to show more than a single parameter at a time - the combination of parameters, leading to vulnerabilities.
  - and therefore also to find a way to aggregate these parameters.
- Spider graphs: in edge-cases spider graphs can result in data not being represented in the right manner (for example, when two axes have the value 0, and one axis a very high value, this will almost not be visible - this can be solved partially by not starting at point 0).
  - Bundling similar properties is another point to look further at, creating a distinguishable “shape” in a spider graph.
  - One suggestion was made that it would more appropriate to use a Venn-diagram.

  - The practitioners felt the user of the TREsPASS model would be given too much freedom if all the parameters were editable directly through the interface. Instead the emphasis should be to be able to select types of actors ('Stepping-stone', 'ostrich', and so on).

- 'The Security Onion': a layered approach to security.
  - Despite investing heavily in their security defences many organisations are still finding their systems regularly compromised. The problem these organisations face is they are focusing too much on the defensive controls at their network perimeter in the false belief that this makes it difficult for their systems to be compromised. However, time and time again we see that once the perimeter controls fail attackers have easy access to the organisation's sensitive assets.
  - The Security Onion is an approach how to secure corporate assets by implementing a multi-layered approach to security incorporating the key cornerstones of people, process and technology. With this multi-layered approach should one security layer fail the other layers can compensate and continue to secure key organisational assets.
  - The "onion" might be interesting from a visualisation point of view, it was suggested.
- Cloud example: physical location of data is either unknown or irrelevant. Access control is interesting, however, and maybe there can be more focus on those aspects.
- The practitioners suggested to highlight the distinction between social-engineered / non social-engineered attack-steps for both the digital attack-steps and the physical attack-steps as we presented. This would mean this would need to be annotated in the attack tree decoration.
- Quantifying can happen within a range. But this range should be shown or explained.
  - On an ordinal scale, what does "high" precisely mean?

Attack paths

- They all doubted if showing all possible attack paths was useful. From a business point of view the top ten attack pathswould be enough.
  - We think a top ten view can be one of the visualisations. This would be the one that you export and take to the next meeting with your supervisor or manager.
  - We think that showing all possible attack paths can somehow be useful if the user is looking for the best or most efficient countermeasure. By showing all countermeasures and indicating their control strength (in keeping the attacker from reaching his goal).
- Minor comments:

  - Why do the radial attack-trace visualisations start from 3 o'clock instead of 12 o'clock?
  - Why do we read the radial attack-trace visualisations counterclockwise?

Building the Attack Navigator Map

- The practitioners panel preferred a visual for the ANM that was like a floor-plan, rather than a more abstract translation of it, although they liked the aesthetics of the latter.
- The colour scheme was described as"very 70s".
- The nested-ness of the digital sphere within all the physical boundaries seemed confusing to the panel. Showing both these realms suggests that it is really hard to get to the digital assets. Whereas there might be ways to get to those digital assets without leaving your own house.
- One suggestion was to split the physical map from the digital map, and make it possible to view those separately.
- A common case was mentioned, what happens if you're not the only tenant in a building, and how the policies connected with this fact are to be represented?
- Using the very same graphical 'Legend', or 'Key', and the same visualisation principles for the ANM, for example, might not be a good idea, since it is natural to read a floor-plan that has thicker wall sections as being more protected, whereas the opposite might be case.
  - In a building.
  - In a datacenter.
  - On a server?
- There was doubt how useful it is to show physical and digital space within a single map. If this is to be the case, they should somehow be represented so that their differences will be apparent.

  The panel was very much interested in this relationship described above, between the physical map and the digital map. They wondered in which ways physical access leads to digital access? Except for very-well encrypted data (where also the keys are not available in the same cloud infrastructure), physical access gives you complete access to digital data. With physical access you can very often boot into root-level access to servers.

Building ANM using the Wizard

- One thing that was missing, according to one of the panelists (Rob Cordes), was the possibility of modelling interpersonal relationships between actors:
  - Process hierarchies.
    - Can these be compared to policies? They describe the way the different departments, roles and actors rely on each other in doing their job within the company, and how do they work together.
    - Similar to BPML, and machine readable, but able to refer to these relationships.
- Alternative approach:
  - Starting point: identify and provide an inventory for the assets.
  - Estimate impacts whereupon the asset was compromised (how much damage would it cause financially and in terms of business reputation, and so on).
    - An alternative is to work outwards, building different types of controls and countermeasures around the assets that have been identified.

Properties - Actors

- We looked at a few properties and agreed to send full list to practitioners for further feedback (pending). Some of their initial feedback was that the use of actor properties such as “age”, “nationality” and “gender” should be explored in greater depth. Having said this, they also had the feeling that in many companies this is a sensitive area. On the other hand, IBM for instance (as well as other US-based organisations), has a list of embargoed countries, hence this could also be a good property to develop.
- The panel suggested “seniority” as an alternative property to investigate.
- “Personality colour” can represent persona character traits (where a colour code can be used to distinguish a particular type of personality) and this is potentially interesting from a visualisation perspective.
- The panel mentioned The CALUWE test[1] where companies evaluate their employees and categorise them by using colours (see personality colour).

[1] https://www.managementdrives.com/ and The CALUWE test

Data

- 'Up-to-date' solutions, an essential but significant challenge.
- Quality: "garbage in, garbage out".
  - certainty: a lot is based upon professional judgement and "guess-culation".
    - One issue is how to acknowledge and communicate this.
- There should be a clear message to the user about the process is: initial modelling, leading to a first analysis, then to visualisation, and thus to iteration of this process.
  - for them it all seemed to happen in the same view, should fix that in UI.
- Some of the practitioners felt we were forgetting about data and the movement/relocating of data within the digital infrastructure. They felt we focused too much on representing the data on one physical place (VM1), while in practice the data is likely dispersed over many servers, VMs and even physical locations.

Determining the dominant narrative: the approach of Risk assessment in *Shell* organisation

1. Determine which assets you most want to protect.
2. What is it's value? Determine the value of an asset based on the possible damage when an attacker gets hold of the asset (in other words, to bring about a state of affairs not desirable for your business).
3. Determine, based on this value, who possibly would get hold to this asset (attacker).
   - The attacker profile plays an important role:
     a) Who wants my assets, and why?
     b) What does the attacker want to achieve?
        i. Define your attacker profile ("know your enemy").

Purpose of the TREsPASS model The practitioners panel had some questions regarding the use of the model. They question the way this system can be updated to do the analysis on a more regular basis.

General feedback / Open questions

- Risk evaluation for the complete practitioner's panel is more based on the possible impact an attack causes or might cause. This is still a very important aspect from a business point of view.
- For the practitioners panel, risk consist of three elements:
  - Impact.
  - Vulnerability.
  - Likelihood.
- Analysis shouldn't be an unknown quantity, or a 'black box'.
  - people should know why specific countermeasures are needed

B. Overview of WP4 Evaluations 2014-6

Total number of participants: **266**

This breaks down as:-

1. LEGO: *174*
2. Paper Prototyping: *60*
3. ANM design concepts and prototypes: *15*
4. Other, including InterActor app: *29*

Time-line and details of principal WP4 engagements

2014

January, TRE$_S$PASS **General Meeting LEGO workshop, Brussels**. 1 hour session. A mixed group of researchers and professionals used the LEGO method to brainstorm the way in which data could be gathered from case studies and incorporated into the design of the TRE$_S$PASS tools. *Participants: 30*

August, **MSc students and alumni Lego workshop**, 3 hour session, held at ISG, RHUL, London. Participants included those who are currently studying for the MSc Information Security at RHUL, and those recently graduated, most of whom are also established practitioners. They were given the opportunity to use LEGO and were able to provide detailed written as well as verbal feedback. *Participants: 6*

December, **Lego session with City IT employee**. 2 hour session, looking at how the technology of the workplace, a financial institution in the City of London, impacted on business efficiency and competitiveness, as well as issues around internal communications and roles. *Participants: 1*

August, TRE$_S$PASS **General Meeting, Copenhagen. 2 x 1 hour LEGO session**. A mixed group of researchers and professionals used the LEGO method to brainstorm the way in which data could be gathered from case studies and incorporated into the design of the TRE$_S$PASS tools. *Participants: 15* (estimated)

May, **People in Security, Centre for Doctoral Training, LEGO and Paper Prototyping workshop, RHUL**. This 3 hour session presented the method in the context of current research in the area, and gave participants the opportunity to see how the methods

dealt with a complex cloud-based scenario. As well as training participants, via practicing it themselves, and detailed verbal feedback was also gathered. *Participants: 15* (estimated)

2015

February, Royal Society, London. Presentation on password visualisation for the ANM, and poster representing current WP4 work on the ANM. Participants: 25

February, **Royal Society, London**. Presentation on password visualisation for the ANM, and poster representing current WP4 work on the ANM. *Participants: 25*

April (28-29th), **Cyber Security and Privacy Innovation Forum (CSP) 2015, paper prototyping workshop, Brussels**. 1 hour session. Participants were invited to use the paper prototype of the Attack Navigator Map produced by LUST, prompted by an initial discussion of a scenario. *Participants: 30* (estimated)

March and June, **ANM Practitioners Panels, LUST**, two meetings, each 1.5 hours (09-03-2015 and 04-06-2015). Both panels were held at LUST presenting the current state of the ANM, the second session with updated elements responding as far as possible to the earlier given in March 2015. Responses were categorised as model related (or general), and visualisation and interface related, or a combination of both. Feedback was gathered and sent TRE$_S$PASS co-ordinators and documented in D4.1.2. *Participants: 8*

May, **Pallion Action Group, LEGO workshop, Sunderland**. 1.5 hour session with low-income participants to model start-up businesses, and examine risks to data associated with the business models proposed, and to look at what kinds of support and connectivity they would require for success. *Participants: 6*

October, **Edith Cowan University Paper, Prototyping workshop, Perth, Australia**. 1.5 hour session.Participants were invited to use the paper prototype of the Attack Navigator Map produced by LUST, in order to conduct an analysis of risk in a specified context. The activity provided an example exploring the physical, digital and social aspects of risk. It promoted the application of attack tree concepts to a situation use case. For detailed feedback see D9.1.6. *Participants: 15*

November, **Direct Payments workshop: Sunderland local government, UK**. 3 hour session. This was the second workshop held in Sunderland, building upon the scope and insights of the first, above. The workshop organisers, TREsPASS partners Royal Holloway, contacted EU living labs member Sunderland City Council and community service provider Pallion Action Group to ask if they could recruit SMEs and service providers from the statutory body to take part in this workshop. 18 participants representing 7 organisations agreed to participate. For detailed feedback see D9.1.6. *Participants: 18*

November, **IASDR 2015, LEGO workshop, Brisbane, Australia**. A full-day workshop using LEGO research methods with a mixed group of researchers, practitioners, and designers. Titled ‘My Place: Designing for Safety and Security in a Digitally-Flecked World’,

was held at the International Association of Societies of Design Research (IASDR). For detailed feedback see D9.1.6. *Participants: 10*

2016

March, **Edith Cowan University, Paper Prototyping workshop**, Perth, Australia. 1.5 hour session. Participants were invited to use the paper prototype of the Attack Navigator Map produced by LUST, in order to conduct an analysis of risk in a specified context. The activity provided an example exploring the physical, digital and social aspects of risk. It promoted the application of attack tree concepts to a situation use case. For detailed feedback see D9.1.6. *Participants: 15* (Report produced that covers both sessions)

May, **Practitioner feedback, LUST, the Hague**, 2 one day sessions where a security practitioner worked side by side with a TRE$_S$PASS consultant, to elicit feedback on the value of the ANM to practitioner job profiles. Report produced and sent to MT. *Participants: 1*

May, **CyberUK in Practice, LEGO workshop**. A hands-on I.25 hour session, run with a diverse range of UK security practitioners, showcasing the method and gaining feedback and commitments to further involvement in testing. The session took as its starting point a scenario in the workplace where an employee is intensively monitored. *Participants: 55*

June, **Summer School, RHUL**. The ANM was demonstrated by a TRE$_S$PASS consultant to security practitioners to gather feedback on the value of the ANM. Report produced and sent to MT. *Participants: 5*

June, **Summer School, RHUL**. Current TRE$_S$PASS LEGO methods were carried out over a period of 3 days with a mixed group of security practitioners, MSc Information Security students, and researchers from other backgrounds, on occasion in conjunction with the ANM, entering data from the physical modelling sessions. Report produced and sent to MT. *Participants: 15*

July-September, **Study Group, RHUL**. TRE$_S$PASS LEGO methods were given to a small group of Information Security Group MSc students (including current practitioners) over two sessions spread over time. In the first session the group created a physical model of a commercial scenario that has been designed in an earlier meeting of the group. Later, they returned in order to use an early iteration of RHUL's app, entering data from the physical model and visualising the data as actor network maps. *Participants: 3 (plus one working digitally in a remote location)*

July-October, **Practitioner panel, RHUL**. Meetings held with a group of senior to mid-level security practitioners who advised on the development of WP4 tools including InterActor, as well as on current working practices and concerns, and offer feedback on TRE$_S$PASS visualisation and data-gathering methods. *Participants: 4*

October, **Practitioner feedback session, GCHQ, Cheltenham, UK**. Demo and feedback from national security agency: People-Centred Security, Sociotechnical Security Group, CESG., and others. *Participants: tbc*

October, **Relating Systems Thinking to Design (RSD5), Toronto**. Workshop with systemic design practitioners and researchers, using the TRESPASS LEGO method and InterActor app in conjunction. *Participants: 30, tbc*

C. Visualisation Competition 2015: material

C.1. Brief Visualisation Competition

As part of TRESPASS, an EU project about information security risks, we are running a competition for visualisations that capture the social and technical complexity of so-called "cyber attacks". We are asking you to think about cyber attacks from one of the following perspectives:

Cyber attacks on people

- Explain: how your personal safety and security might be threatened by an attack on a piece of technology. What form does this attack take? How might it be experienced? What makes this attack successful?
- Innovate: an idea for a new social media app that can tell you if a cyber attack is affecting you. What would make this social media app effective and reliable? Why would people want to use this app?

Cyber attacks on the State

- Explain: how the security of the State might be threatened by an attack on its digital and cyber infrastructure. What form might an attack take? What are the impacts of such an attack? Who wins and who loses with such an attack?
- Question and Discuss: What motivates the political discussion about cyber attacks? Can we separate attacks in cyber space from behaviours and attitudes in physical space? Are cyber attacks on the State political or technical attacks?

Cyber attacks on technology

- Explain: how the security of technology and technologically held data might be threated by an attack on the technology? What form might an attack take? What might make this attack successful?
- Share: brainstorm with others to explain how such attacks are formed. Devise a game or a puzzle to collaborate with others to deepen an understanding of such attacks. Innovate a crowdsourcing idea to find new ways to gather and share information about such attacks.

We are interested in visualisations that focus on one of the perspectives described above. We are inviting participants to submit a project that develops a visualisation that depicts the nuances of information sharing, vulnerability and risk in an everyday world where the social and technical entwine.

Participants are invited to design and submit a project in one of the two categories below. The project may be either a one-page static visualisation, wireframe or storyboard or a scientific poster (in paper or digital form).

Projects in each category will be judged by a panel of experts in visualisation and we shall produce a short production run of the winning projects and make them available in a range of venues, both virtual and physical, published under the Creative Commons licensing mechanism. As we sign-up venues, these will be posted to this website.

All projects must be visually eye-catching and informative. All projects must in some way critique or question generally accepted principles or beliefs about cyber security risks and the way they are visualised.

Category One: In this category, select one of the cyber security perspectives listed above and produce a data visualisation, wireframe for an interactive tool or narrative, or a storyboard narrative to explore the chosen perspective, developing a narrative that encourages interaction with the viewer. The project (submitted as a one-page PDF or printed poster) will explore how the social meets the technical in a particular attack whilst questioning generally held principles about how these attacks are formed and how they succeed or fail, and also how they can be visualised.

Category Two: Scientific Poster: In this category, select one of the cyber security perspectives listed above and develop a poster presentation to present research information in the area of socio-technical risk from one of these perspectives. It may be produced in paper or digital form. As with Category One, the poster must encourage interaction with the viewer and explore how the social meets the technical in a particular attack. Regardless of the perspective selected, the presentation of the scientific results must both inform and question some of the generally held principles and beliefs about how these attacks form and succeed. These posters are typically still designed to be eye-catching, but also complex and layered, consisting of images and text.

C.1.1. Jury of the TRESPASS visualisation competition

Ben Fry is principal of Fathom, a design and software consultancy located in Boston. He received his doctoral degree from the Aesthetics + Computation Group at the MIT Media Laboratory, where his research focused on combining fields such as computer science, statistics, graphic design, and data visualization as a means for understanding information. After completing his thesis, he spent time developing tools for visualization of genetic data as a postdoc with Eric Lander at the Eli & Edythe L. Broad Insitute of MIT &

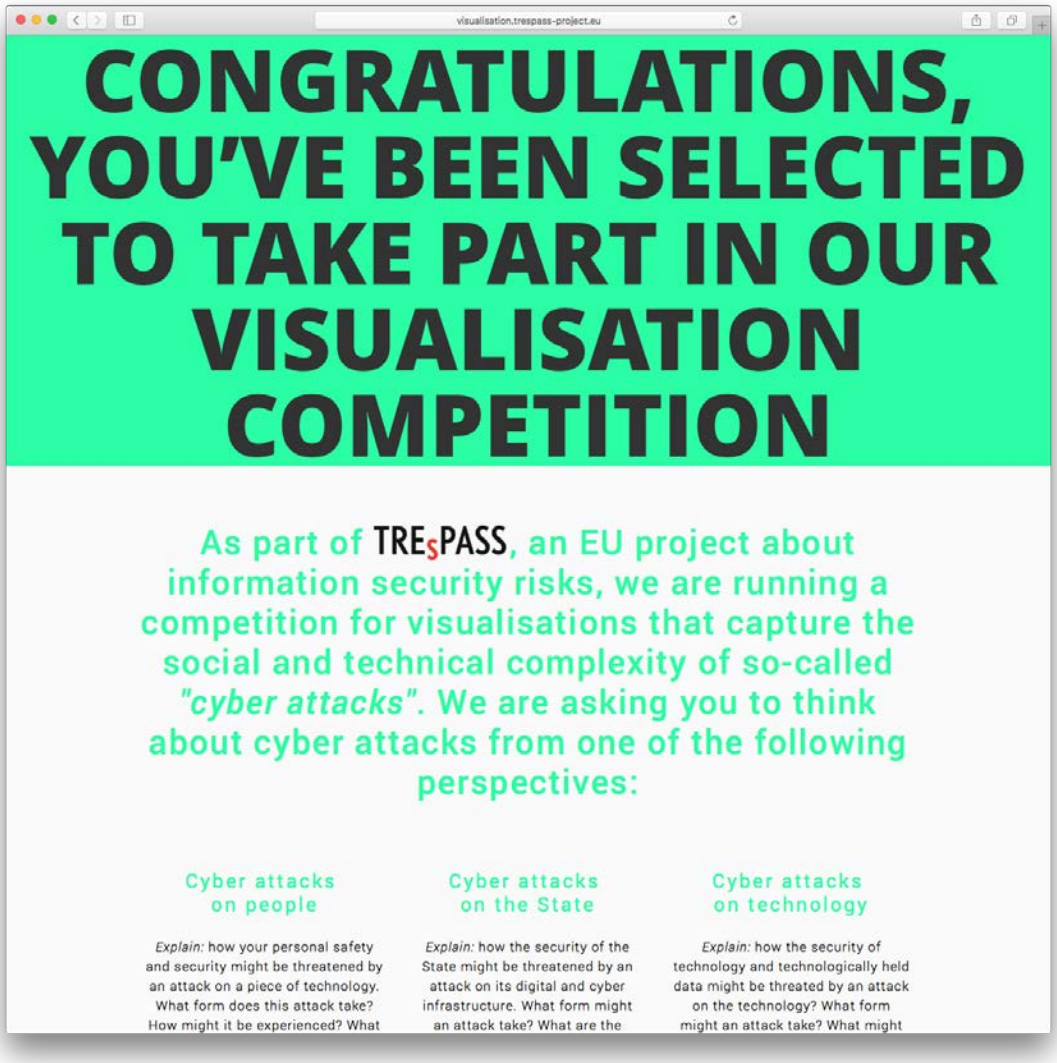


Figure C.1.: Screenshot of the TRESPASS visualisation competition micro site

Harvard. During the 2006-2007 school year, Ben was the Nierenberg Chair of Design for the Carnegie Mellon School of Design. At the end of 2007, he finished writing Visualizing Data for O'Reilly. In 2011, he won the National Design Award for Interaction Design from the Cooper-Hewitt. With Casey Reas of UCLA, he currently develops Processing, an open source programming environment for teaching computational design and sketching interactive media software that won a Golden Nica from the Prix Ars Electronica in 2005.

Claude Heath is a researcher in the Information Security Group at Royal Holloway, London, and an artist with an international track record. He was awarded a PhD in Computer Science and Electronic Engineering, at Queen Mary, London, devising a way of drawing shared spaces observed in video of collaborative interactions. His artworks have included drawings, paintings and installations based on the sense of touch alone, and drawings

made freehand with 3d software that emulates the sense of touch. Claude Heath has had numerous solo shows including Schaustelle at the Pinakothek der Moderne, Munich; Fruehsorge Contemporary, Berlin; The Henry Moore Institute/Leeds City Art Gallery, and he has participated in many group exhibitions including at Saatchi Gallery, London, and Kunsthalle Baden-Baden. His work is in collections including The British Museum, Deutsche Bank, Museum Kunst Palaast, Dusseldorf, and Pinakothek der Moderne, Munich.

Lorraine Gamman is Professor of Design at the University of the Arts London; She founded the award-winning Design Against Crime Research Centre at Central Saint Martins in 1999 that has delivered many student projects and been supported by external funding between 2000-2015 from the AHRC, ESCRC , EPSRC and the Design Council/Home Office, to deliver numerous academic outputs. She is currently (2014-16) Principal Investigator on the AHRC funded 'Extending Empathy' network 2014-15; Co-Investigator on the EU FP7 funded 'Graffolution' project and the Danish "Social Games Against Crime" project funded by TrygFonden; also Principal Investigator on the AHRC-funded 2015-16 'Design Thinking for Prison Industries' project that will explore how best to co-design products against crime with prison inmates in the UK and India to help visualise the idea of anti crime design as a form of restorative justice in action. Gamman has published widely on design and visual culture, has produced numerous co-authored design articles as well as journals. Books include Gone Shopping – the Story of Shirley Pitts, Queen of Thieves, Bloomsbury, 2012.

Manuel Lima is the founder of VisualComplexity.com, Design Lead of Codecademy, and a regular teacher of data visualization at Parsons School of Design. He is a Fellow of the Royal Society of Arts, nominated by Creativity magazine as "one of the 50 most creative and influential minds of 2009". Manuel is a leading voice on information visualization and has spoken in numerous conferences, schools and festivals around the world, including TED, Lift, OFFF, Eyeo, Ars Electronica, National Academy of Sciences, Harvard, MIT, Royal College of Art, NYU Tisch School of the Arts, ENSAD Paris, University of Amsterdam, MediaLab Prado Madrid. His first book Visual Complexity: Mapping patterns of information has been translated into French, Chinese, and Japanese. His latest The Book of Trees: Visualizing Branches of Knowledge, published in April 2014 by Princeton Architectural Press, covers over 800 years of human culture through the lens of the tree figure, from its entrenched roots in religious medieval exegesis to its contemporary, secular digital themes.

Raffael Marty is one of the world's most recognised authorities on security data analytics and visualisation. Raffy is the founder and CEO of pixlcloud, a next generation visual analytics platform. With a track record at companies including IBM Research and ArcSight, he is thoroughly familiar with established practices and emerging trends in big data analytics. He has served as Chief Security Strategist with Splunk and was a co-founder of Loggly, a cloud-based log management solution. Author of Applied Security Visualisation and frequent speaker at academic and industry events, Raffy is a leading thinker and advocate of visualisation for unlocking data insights. For more than 14 years, Raffy has worked in the security and log management space to help Fortune 500 companies defend themselves against sophisticated adversaries and has trained organisations around the

world in the art of data visualisation for security. Zen meditation has become an important part of Raffy's life, sometimes leading to insights not in data but in life.

The full micro site can be found at www.visualisation.trespass-project.eu.

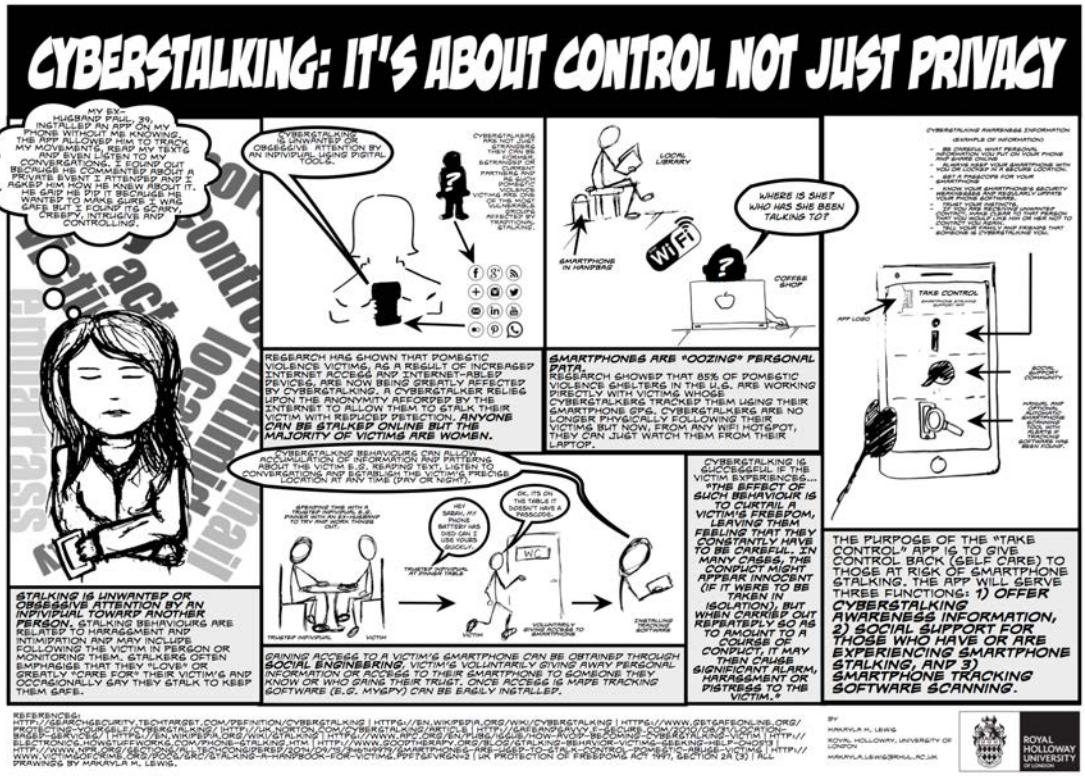
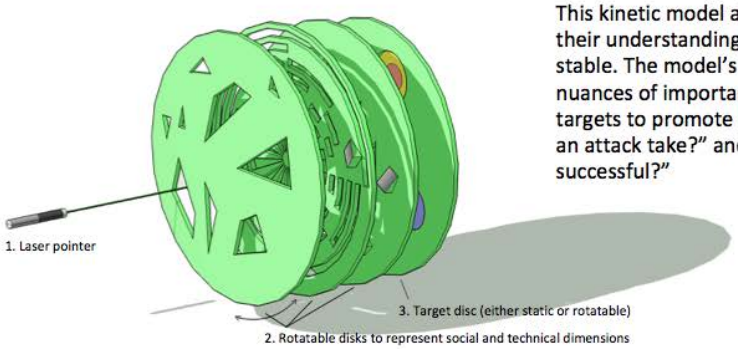


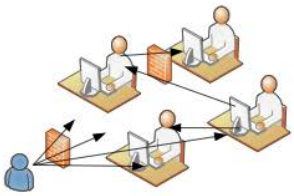
Figure C.2.: First prize winning poster by Makayla Lewis: *Cyberstalking: its about control, not only privacy!*

Figure C.3.: Second prize winning poster by Bente Brunia: *Hacking has never been easier, select your personal virus now!*

Kinetic Visualization of the Social and Technical Complexity of Cyber-attacks



This kinetic model aims to support users in advancing their understanding of cyber-attacks as anything but stable. The model's dynamic nature depicts multiple nuances of important cybersecurity concepts and targets to promote discussions like "What form might an attack take?" and "What might make an attack successful?"

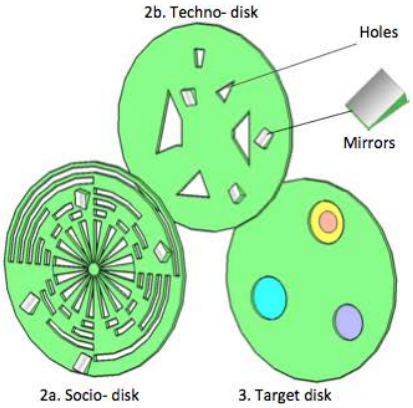


Communicating methods of multi-staged attacks on socio-technical systems to non-experts is not trivial. Illustrations like a Swiss cheese model lend themselves well for initial explanations how a hazard can pass through a system and cause losses. However, such illustrations can be misguiding due to their static nature. One could overlook that the exploitation of vulnerabilities in socio-technical systems is an interactive multi-step process. For instance, advanced persistent threat sources can establish a foothold and then 'step back' to a previous layer of the system. A tangible kinetic model can illustrate such attack methods in more detail and support users in contemplating and discussing how attacks can unravel.

The kinetic model presented here exhibits a continuously changing socio-technical system, which can deflect or channel malicious hidden attacks. It highlights a commonly under-emphasized idea that established footholds can be used for diverting attack vectors. *It therefore challenges the monotonicity assumption, when the attackers never needs to backtravel*. Formed as an interactive puzzle, the tool is a perfect enabler of and mediator for discussions between an expert and a non-expert, who can explore the nature of complex systems and cyber-security herself. By inviting the user to find out successful attack paths, the model serves as a boundary object to construct advanced understanding of security concepts.

Several components constitute the model:

1. Laser pointer. A threat, as a source of impending danger, is embodied as a light. The laser beam exemplifies the invisible nature of attacks, while also enables more advanced illustrations. A shimmering light may indicate instants when the threat manifests itself. A moveable pointer can illustrate different attack vectors. Additionally, a set of pointer locations can be pre-defined to enhance the model.
2. Rotatable discs illustrate the social and technical dimensions of the system. The social disk equipped with holes and mirrors demonstrate that social engineered attacks can go through the plane or the attack can be reflected to another layer. The technological disks illustrate different security layers and perimeters. The model may include several technical or social layers. Specific elements characterize properties of the system: holes illustrate vulnerabilities, deflectors (mirrors) point out that attacks can step back, and non-transparent parts can assist in detected the attacks.
3. Target disk. This plane describes assets as targets of attacks. With more than one sector characterizing a particular asset, the model highlights that results of attacks can differ (for instance, an attack can results in either destruction or degradation of a service). Besides, partially successful attacks (when the attacker obtains user-level access to a system, but not administrator rights) can also be illustrated.



This simple yet comprehensive model of the complexity of cyber-attacks can be easily implemented and extended. The planes can be made using a 3D printer. A smoke machine or spray can assist in tracing the laser beam. Step motors and controlled lasers pointers can convert the interactive puzzle into a mechanical art installation.

All in all, this kinetic visualization model combines clearly understandable concepts of vulnerabilities and movable social and technical planes. By demonstrating interactions between these components, the model aims to assist users in developing and discussing insights related to essential cybersecurity aspects.

This one-pager describes an interactive tool (Category One) to illustrate Cyber attacks on technology (Perspective three) of the Trespass visualization challenge. The model was made using "green laser pointer" and "24 degree segment transducer" open models from 3D SketchUp Warehouse.

Figure C.4.: Second prize winning poster by AlexOnline: *Kinetic Visualization of the Social and Technical Complexity of Cyber Attacks*

D. Advanced visualisation workshop 2016: outcomes

D.1. Outcomes of the workshop

The participants of the advanced visualisation workshop worked in five different groups with data from the ATM case study, the geographical data set on ATMs in Lisbon, their attacks and all kinds of social data around this. Here we present some of the results. All results are also published on visualisation.trespass-project.eu.

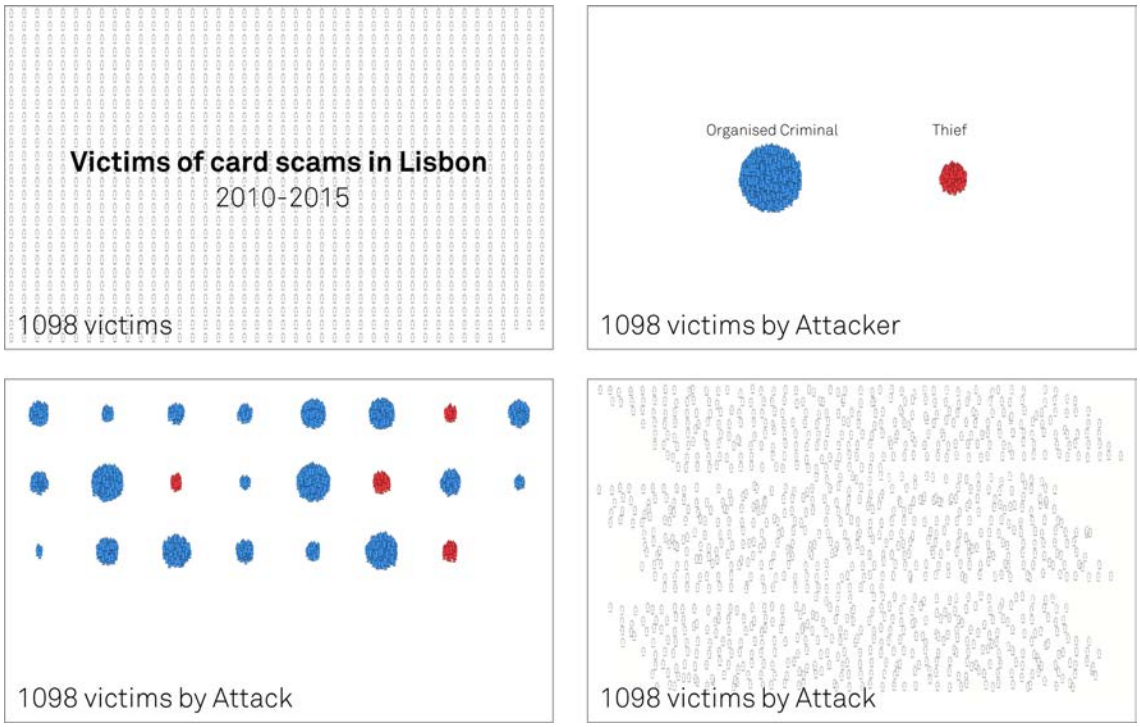


Figure D.1.: Four screenshots of an animation based on the ATM data set. This group came up with a narrative that tried to make the attack personal again. From the data, they could calculate the number of victims per attacked ATM. They visualised each victim as a person, and showed how many were victimised by organised crime or just common thieves, as well as how many victims were made per ATM.

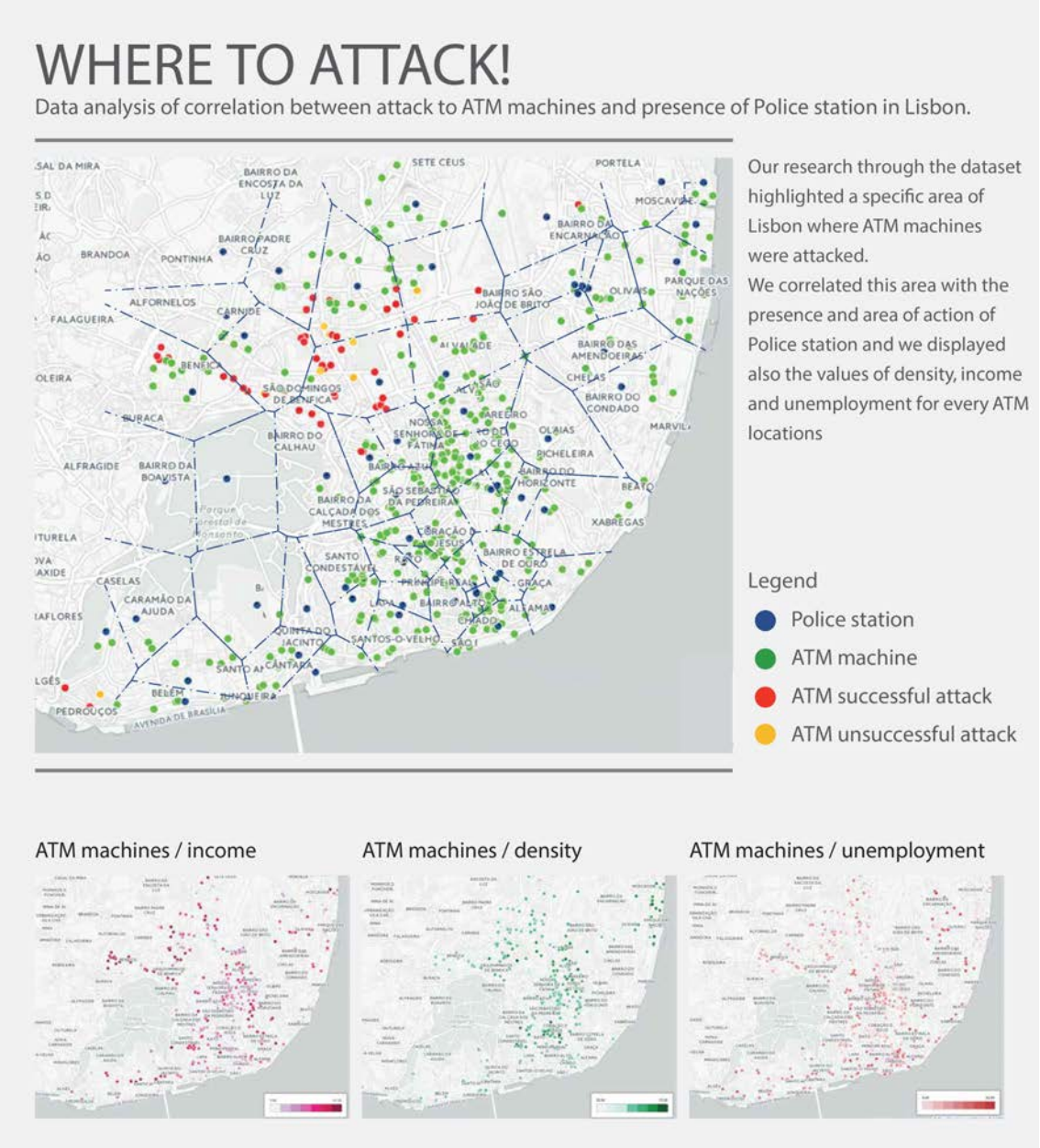


Figure D.2.: This group focused on where attacks were taking place and if there was a relation with external factor, as income in the neighbourhood, population density, and so on.

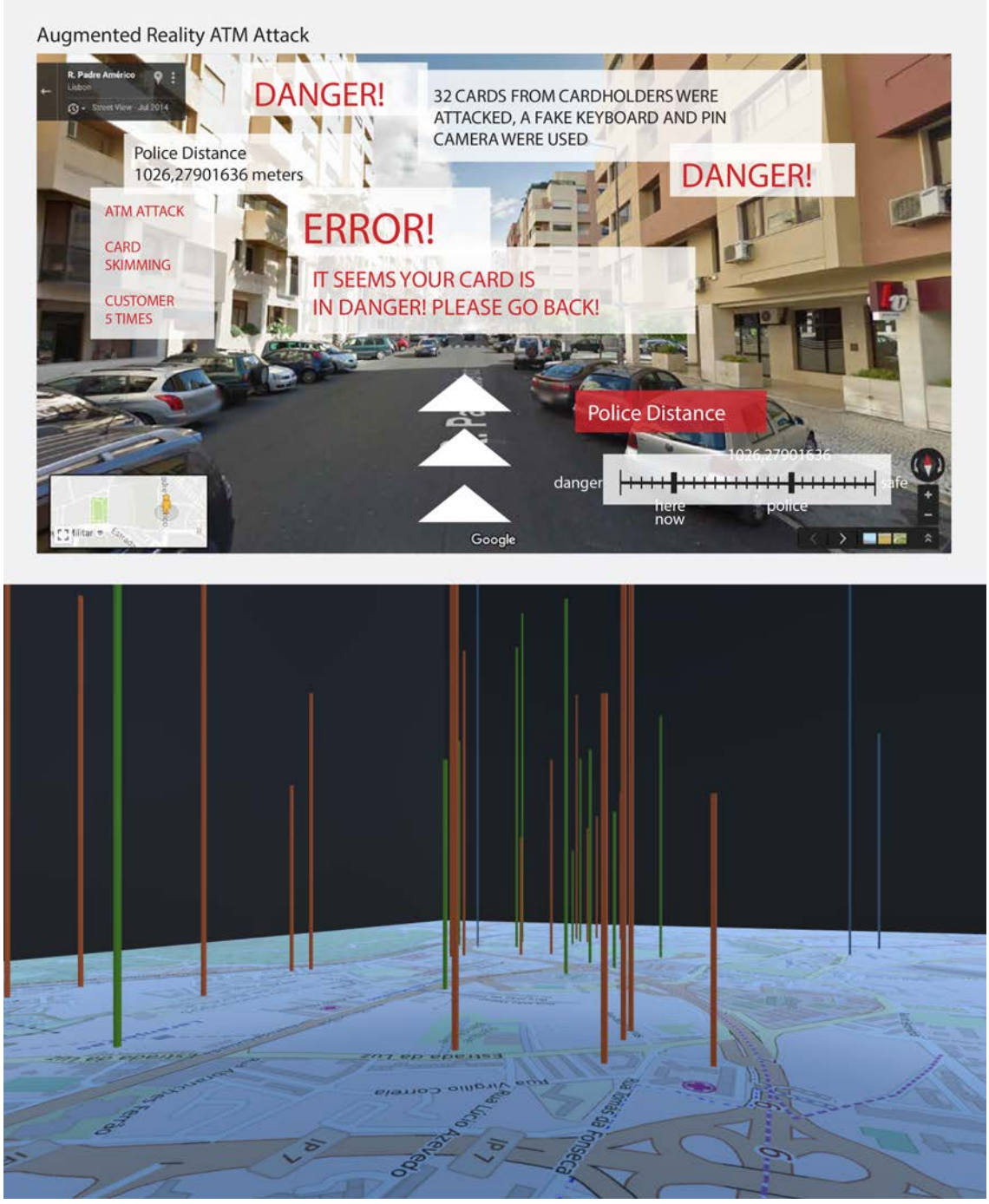


Figure D.3.: This group presents an alternative view on vulnerability information, by putting it back on the streets. The bottom image represents a game-like first person visualisation of various ATM types.

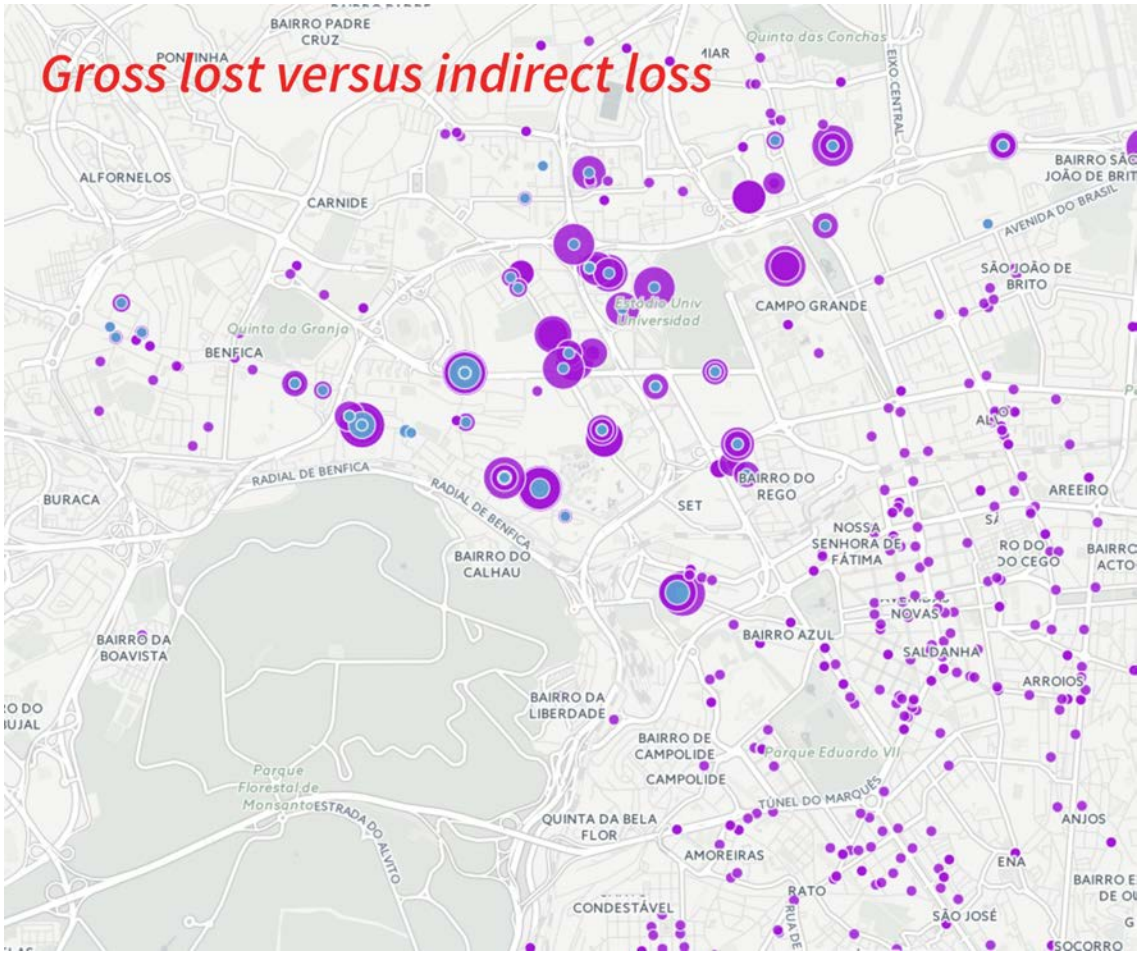


Figure D.4.: This group investigated the gross loss versus the indirect loss and first plotted this on a map.

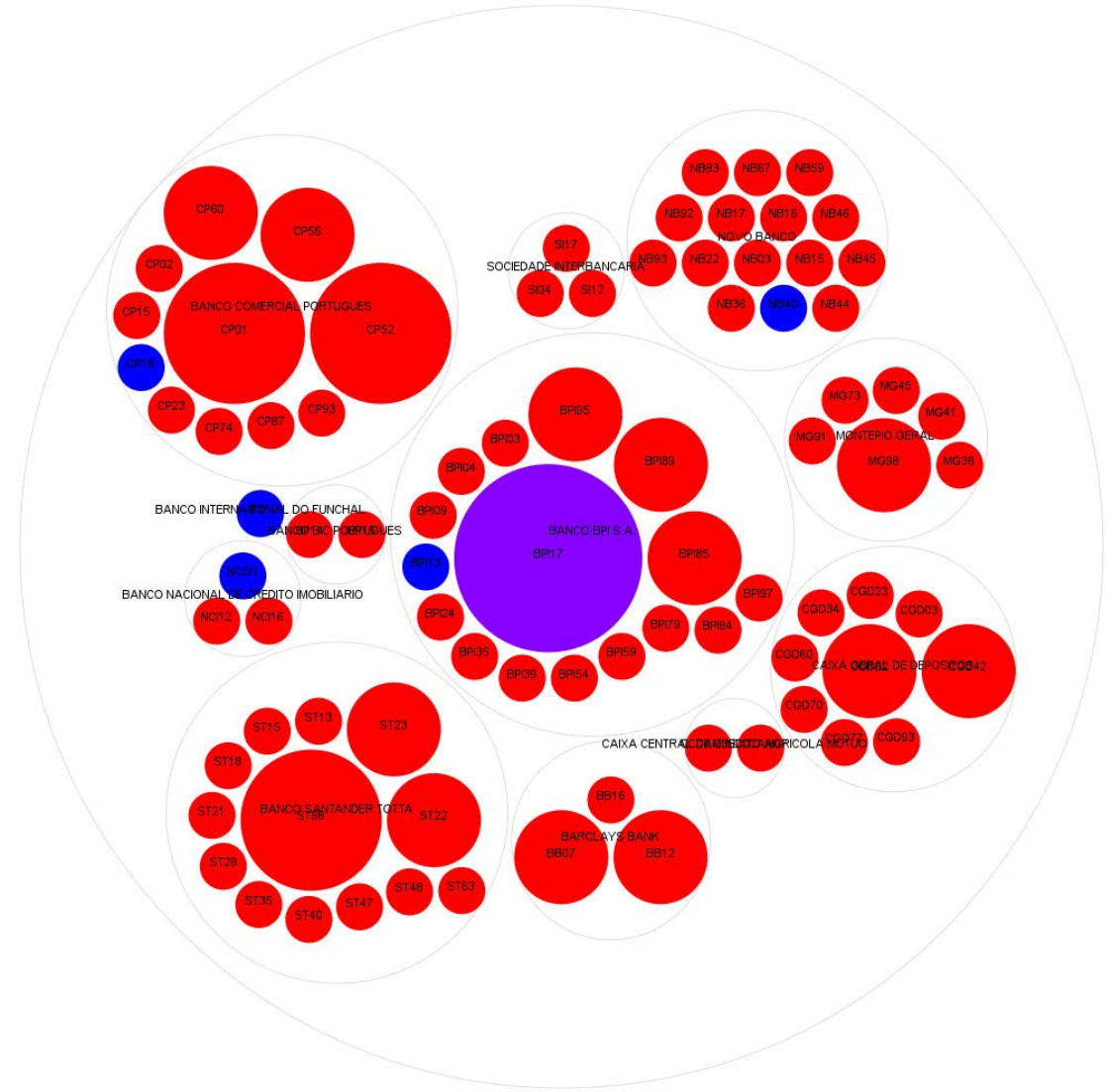


Figure D.5.: A further exploration as result from the research what was found in Fig. D.4, it became clear that the geographical aspect was not the most important aspect to visualise. In this figure, the loss is visualised per bank, with differentiation between manual and logical attacks.

E. Complexity of attack trees

E.1. Introduction

Automated generation of modelling languages based on trees is indeed not new. In the area of reliability and safety several approaches aim to automatically generate *fault trees* from system models (Liggesmeyer & Rothfelder, 1998; Majdara & Wakabayashi, 2009; Chen, Avrunin, Clarke, & Osterweil, 2006). Also recently, Vigo et al. developed a method that infers attack trees from a process algebra specification (Vigo, Nielson, & Nielson, 2014). A problem that has not been fully addressed yet in the topic of automatic generation of attack trees is complexity, where complexity is regarded as how humans create and understand attack trees. The abstraction power of the human brain, makes us to believe that complexity is related to the size of the tree. We thus set up a simple preliminary study on how humans understand and create attack trees. Our hypothesis is that the smaller the size of the tree the better comprehensible it becomes for humans.

We evaluate comprehensibility in terms of completeness and human-abstraction. Completeness refers to the ability of an attack tree to describe a security scenario. Human-abstraction, instead, is a property reflecting the structural difference between an attack tree generated by a human and another semantically equivalent attack tree.

E.2. Interviewees: a phenomenological approach

We address the above challenge by means of individual interviews. We trained a group of nineteen students at the University of Luxembourg in the attack tree methodology. The interviewees were enrolled students at the University of Luxembourg in the Master of Information and Computer Sciences. At the beginning of the interview the interviewer explained to the participants the main idea behind the research. Participants were also familiarised with the Attack-Defense Tree Tool (ADTool)[1] by means of a short tutorial.

Each interviewee was provided with a very short description of an attack scenario, that is, how an attacker can steal money from an online bank account. In addition, we gave them a list of keywords specifically related to the attack scenario, namely “Phishing”, “Pharming”, “Software Keylogger”, and “Hardware Keylogger”, even though we were not expecting the interviewees to be familiar with all these keywords.

[1] http://satoss.uni.lu/members/piotr/adtool/

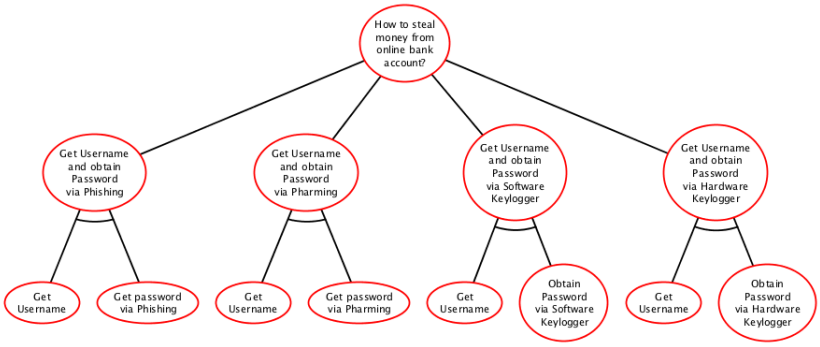


Figure E.1.: An attack tree in normal form.

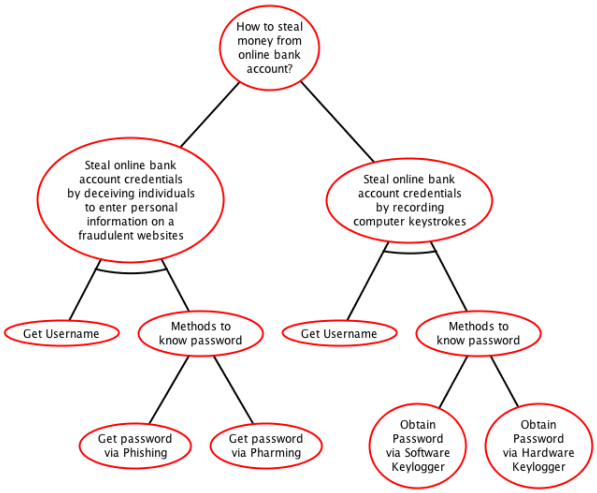


Figure E.2.: An attack tree with fewer categories than the one in Figure E.1.

E.3. The questionnaires

In order to focus on the tree structure and its communication power, we designed three structurally different, but semantically equivalent, attack trees (see Figures E.1, E.2, and E.3). Note that, they all capture the same set of attack vectors.

Based on this security scenario, we split the group of nineteen interviewees in two, and perform two different type of interviews. The first one was aimed at evaluating the communication power of the attack tree, while the later was aimed at measuring the mismatch between the abstraction-power provided by the attack tree and the abstraction-power of humans.

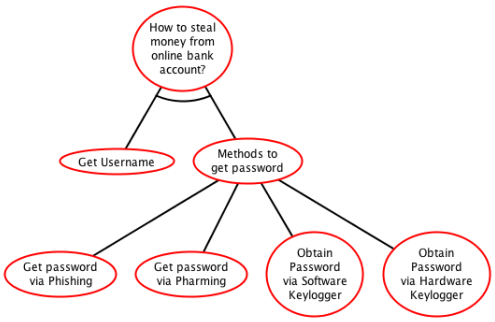


Figure E.3.: An optimised attack tree in terms of number of leaf nodes.

E.3.1. Evaluating completeness of the attack tree.

In order to evaluate this variable, we confronted thirteen interviewees with the attack trees depicted in Figures E.1, E.2 and E.3, in a random order. The interviewee was later asked to choose the attack tree which explains better the attack scenario he/she had in mind. It turns out that most interviewees picked Figure E.2 as the most descriptive attack tree.

E.3.2. Evaluating human-abstraction of the attack tree.

To the other six interviewees we asked first to generate their own attack tree considering the previously mentioned goal as the root of the attack tree. The task of the interviewer was to measure the time taken by the interviewee to create the attack tree. The interviewer was not allowed to give hints to the participants or interfere with the process of generating attack trees. The interviewer was only allowed to support the participants if they had any technical issues with the ADTool.

Immediately after completing the attack tree, the interviewee was asked to explain the logic behind their design choices. Finally, the attack trees depicted in Figures E.1, E.2 and E.3 were revealed in a random order. The interviewee was asked to choose the one closer to his/her attack tree in terms of structure and content.

Unfortunately, these interviews didn't arise any statistically significant result. We can only make a weak claim stating that it seems that the tree structure used in Figure E.1 is hard to find in a human-generated attack tree.

E.4. Conclusions

This preliminary study doesn't contradict the initial hypothesis that, the smaller the size of the tree the better comprehensible it becomes for humans. However, it doesn't provide answers on how optimised the attack tree should be. Further research needs to be done on this direction.