predict
prioritise
prevent

TRESPASS

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

Deliverable D2.1.2

Final requirements for the data management process

Project:	TRE$_S$PASS
Project Number:	ICT-318003
Deliverable:	D2.1.2
Title:	Final requirements for the data management process
Version:	1.0
Confidentiality:	Public
Editor:	S. Saraiva
Cont. Authors:	S. Saraiva, F. Reis, M. Fraile
Date:	2015-10-30

Part of the Seventh Framework Programme
Funded by the EC-DG CONNECT

Members of the TRESPASS Consortium

1.	University of Twente	UT	The Netherlands
2.	Technical University of Denmark	DTU	Denmark
3.	Cybernetica	CYB	Estonia
4.	GMV Portugal	GMVP	Portugal
5.	GMV Spain	GMVS	Spain
6.	Royal Holloway University of London	RHUL	United Kingdom
7.	itrust consulting	ITR	Luxembourg
8.	Goethe University Frankfurt	GUF	Germany
9.	IBM Research	IBM	Switzerland
10.	Delft University of Technology	TUD	The Netherlands
11.	Hamburg University of Technology	TUHH	Germany
12.	University of Luxembourg	UL	Luxembourg
13.	Aalborg University	AAU	Denmark
14.	Consult Hyperion	CHYP	United Kingdom
15.	BizzDesign	BD	The Netherlands
16.	Deloitte	DELO	The Netherlands
17.	Lust	LUST	The Netherlands

Document History

Authors		
Partner	Name	Chapters
GMVP	Sérgio Saraiva	ALL
GMVP	Fátima Reis	ALL
GMVS	Marlon Fraile	ALL

Quality assurance		
Role	Name	Date
Editor	Sérgio Saraiva	2015-10-30
Reviewer	Mads Chr. Olesen	2015-10-30
Reviewer	Dan Ionita	2015-10-30
Task leader	Sérgio Saraiva	2015-10-30
WP leader	Michael Osborne	2015-10-30
Coordinator	Pieter Hartel	2015-10-30

Circulation	
Recipient	Date of submission
Project Partners	2015-10-30
European Commission	2015-10-30

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

Contents

List of Figures iv

List of Tables v

Management Summary vii

1. Introduction 1
- 1.1. Overview . . . 1
- 1.2. Objectives . . . 4
- 1.3. Foreground and background . . . 4
- 1.4. State of the art . . . 4
- 1.5. Approach . . . 5

2. Data management process 7
- 2.1. Project progress on the topic of data extraction . . . 7
- 2.2. Revision of the initial requirements . . . 7
  - 2.2.1. General baseline of data requirements . . . 7
  - 2.2.2. Refinement of general requirements . . . 9
- 2.3. Requirements for the extraction prototype . . . 9
  - 2.3.1. Required data sources . . . 10
  - 2.3.2. Required data types . . . 11
  - 2.3.3. Representing uncertainty . . . 13
  - 2.3.4. Attacker behavioural patterns . . . 14
  - 2.3.5. Data design patterns . . . 15
  - 2.3.6. Dealing with extensions - a practical example . . . 17
  - 2.3.7. Creating a TRE$_S$PASS data model ready to be extended . . . 21
  - 2.3.8. Cost of the data and cost benefit analysis . . . 23

3. Integration of social and technical data 26
- 3.1. Project progress on the topic of data consolidation . . . 27
- 3.2. Understanding the integration process . . . 28
  - 3.2.1. A practical example - year 3 ATM case study . . . 28
  - 3.2.2. Data definitions . . . 30
  - 3.2.3. Consolidating data . . . 31
- 3.3. Revision of the initial requirements . . . 33
  - 3.3.1. General baseline of consolidation requirements . . . 33
  - 3.3.2. Refinement of the general consolidation requirements . . . 33

3.4. Intermediate requirements on data consolidation . . . 34
3.4.1. Consolidation between domains . . . 34
3.4.2. Collecting and storing the data . . . 35
3.4.3. Data formats and access . . . 36
3.4.4. Consolidation process . . . 37

4. Handling external data sources **38**
4.1. Progress on handling external data sources . . . 38
4.2. Revision of the initial requirements . . . 39
4.2.1. General baseline of integration requirements . . . 39
4.2.2. Refinement of general integration requirements . . . 40
4.3. Requirements on actor integration . . . 40
4.4. Interactions with other WPs . . . 41
4.4.1. Data schema definition . . . 42
4.5. Interaction with external sources of information - project progress . . . 43

5. Future developments and planning **46**
5.1. Next steps . . . 46
5.2. Schedule . . . 46
5.3. Risks and uncertainty . . . 47

6. WP2 Requirements List **48**
6.1. Requirements from WP2 to other WP's . . . 48
6.2. Requirements from other WP's to WP2 . . . 49

7. Conclusions **55**

A. Project Summary **56**
A.1. Case Studies . . . 57
A.2. Overview of TRESPASS Integration . . . 58

List of Figures

1.1. WP2 Tasks Overview: T2.1 Raises and Identifies Data Needs 2
1.2. TresPass Integration Diagram 3
1.3. WP2 Data Management Requirements Organization 6

2.1. Data hierarchy model from TRESPASS base concepts to specific TRESPASS implementations 22
2.2. Data ROI on data protection and prevention 24

3.1. Data hierarchy model 28
3.2. Integration model 32

A.1. Legend for the Integration diagram in Figure A.2. 59
A.2. Integration diagram for the TRESPASS project. 60

List of Tables

2.1. Summary of required data 11

Management Summary

This document presents the evolution in terms of requirements definition within the scope of WP2. We refine the initial requirements and bring up new important requirements for the data management process.

Since initial requirements specification (?, ?), several tasks have been performed, either on the technical extraction prototype development (?, ?) or on the extraction of social data (?, ?). This work has lead to the necessity of detailing the technical and social data consolidation requirements. So far both contexts have been treated separately. However the different contexts of technical and social data determines that this consolidation needs to occur to be able to bring together different semantics and the requirements for that have been identified.

The definition of a data integration model that is able to cope with these different contexts and define ways to represent and transform data from different contexts into a global common schema. However, local adaptations should always be available as different cases might be applicable and seen in the case studies.

Key takeaways:

- Technical and social data consolidation are central processes to be able to accommodate different conceptual types of data
- The data integration model should be defined by a global schema, the sources schemas and the mappings between the sources and the global schema
- Data extraction and transformation is defined between global schema and the source schemas

1. Introduction

Managing requirements, encompasses those tasks that go into determining the needs or conditions to meet the TRESPASS project, taking account of the possibly conflicting requirements of the various stakeholders. Requirements management is a process more than a document that includes gathering (eliciting), analysing (clear, complete, consistent and unambiguous), documenting (recording) and validating (through measurable goals) the requirements.

Concerning stakeholders, an important aspect to consider are future TRESPASS system final users, which include organizations CSO's (chief security officers) and security authorities officials, which can use the TRESPASS system to support security planning and operations. On the other side, another potential user for the TRESPASS system are security consultants that rely on the TRESPASS system to leverage their business offer through the TRESPASS system innovative abilities. Requirements gathering will take these stakeholders into consideration, along with internal project stakeholders (the other WP's).

Requirements management is critical to the success of TRESPASS. The requirements must be documented, actionable, measurable, testable, traceable, related to identified needs or opportunities, and defined to a level of detail sufficient for the TRESPASS system design.

1.1. Overview

The role of WP2 within TRESPASS is to identify structures and processes required for achieving effective handling of data in support to risk management. This includes gathering, extracting, processing and transforming of information from organisation infrastructure as well as from public sources and studies of human behaviour. Therefore, the goal of WP2 - Data Management Process, is to identify and collect data from different sources, gather context specific attack profiles (i.e. historical attacks, known vulnerabilities, etc.), add value to the data collected by interpreting, transforming, analysing, aggregating, validating, classifying and delivering (D2.1.1, D2.1.2) it into a model that complies with other WP's information formats and concepts (WP1, WP6), and information needs for analysis (D2.1.1, WP3).

WP1 introduces the concept of data model, which will be used by WP2 to output data. WP2 will also need to store and process data and for that propose internal data models must be created to store and structure data. In this deliverable we introduce the notion of

separating the base concepts (actor, location, access policies, etc.), from the semantics (rules applied to the data according to the specific contexts, case studies and goals) in order to support pre analysis capabilities, which for instance include performing initial data filtering, prioritization and data inference.

The focus of this deliverable is the revision and finalization of the initial requirements concerning the data management process (?, ?). More specifically, this deliverable revises and extends the set of defined requirements based on research and case analyses performed since month 6 to month 30 of the project.

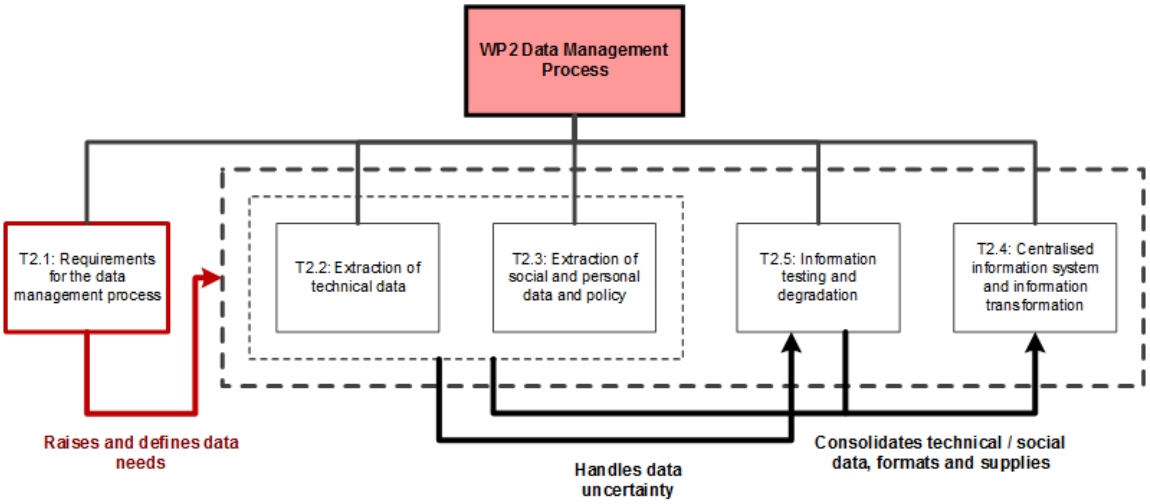

Figure 1.1.: WP2 Tasks Overview: T2.1 Raises and Identifies Data Needs

The first revision of the requirements for the data management process (?, ?), defines the first approach to data management processes: discovery, extraction, categorisation, evaluation and output. After the first approach to prototype technical data extraction, we extend the approach to further steps. The latest form that has been adopted comprises of:

- Discovery - Is the process of exploration of an environment in order to identify the range of available data. It enables to discover potential data sources
- Extraction - Is the process of getting data suitable for supporting risk calculations from an organisation
- Analysis and integrity checking - Is the process of categorisation, transformation, verification and evaluation of the data. Consolidation also fits here
- Storage - The process of defining the data format and store it properly
- Output - How the data will be shared or exchanged with other parties

The focus of the first deliverable D2.1.1 (?, ?) was on specifying requirements for the discovery and extraction step. During the development of the first prototype, further investigation came along regarding analysis and integrity checking and output. In the

current deliverable we have a stronger focus on data consolidation, storage and output and put along aligned requirements.

The data management processes gather data derived from sources which are applicable beyond an individual case (global data), for example from public databases of security information, as well as data relating only to a particular environment (case-specific data). Based on the requirements specification of the navigator map and attacker models, this data is then transformed into the appropriate format for input into the different elements of the modelling process.

The following picture represents the components that are currently involved in TRE$_S$PASS trough the stages: data collection, model creation, analysis and visualisation. The data management framework must supports the information life cycle associated to the whole process.

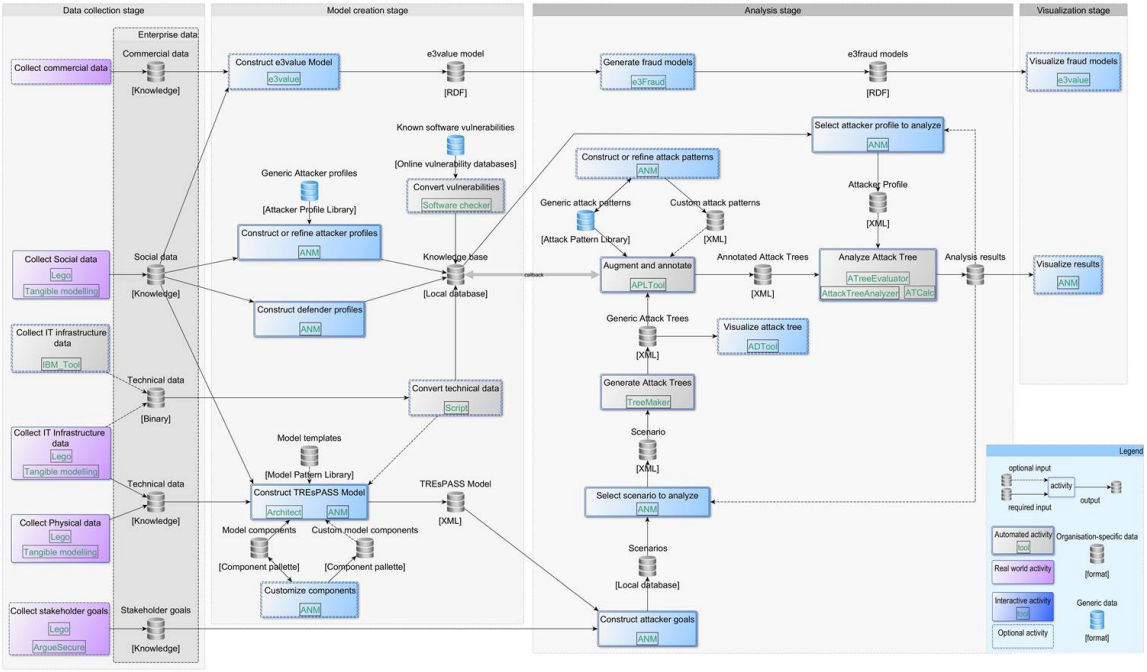

Figure 1.2.: TresPass Integration Diagram

Under normal circumstances, attacker models will typically form part of the global data. While different attackers may been drawn to different environments, according to their interests and skill sets, it is unlikely that any individual type of attacker will be drawn only to a single specific case. The navigator map will be specific to an individual case (although some elements may be reusable, as discussed in relation to model sharing and reuse in the TRE$_S$PASS project (?, ?). When constructing a navigator map for the case, the user can draw on the data gathered from the case environment, in order to support the model development process.

1.2. Objectives

The main objective of this deliverable is to revise initial requirements concerning the data management process. We identify the following sub-objectives:

- Review initial requirements outlined in D2.1.1 (?, ?)
- Extend the requirements based on recent project developments
- Introduce requirements suitable for the data consolidation

The initial requirements outlined in D2.1.1 (?, ?) are referred to in the present document as in the original document (DR2.[N]) while the newly identified requirements are identified as NR2.[N], with [N] being the number of the requirement. Requirements marked 'NR' may be either new requirements or requirements derived from the initial requirements.

1.3. Foreground and background

This deliverable represents incremental work. As the background, we use the initial requirements for the data management process (?, ?). As the foreground, we present revised and new requirements.

1.4. State of the art

State of the art knowledge is available regarding the extraction of data from digital sources, as applied to information security. Intrusion detection and prevention systems (IDS, IPS) collect traffic information to identify any evidence of an attack; security information and event management (SIEM) tools are used as central databases for logs to find any attack evidence (?, ?). Security systems extract data from different digital sources to support organisational decisions. Also, there are other examples in digital environments, outside the security world, such as Business Intelligence systems or ETL (Extract, Transform and Load) techniques (?, ?). If the environment is not digital, data extraction from social or physical environments typically come from audit reports, compliance checking, personnel records and interviews, based on previous security experiences, without objective grounding in data extraction processes. In this item TRE$_S$PASS can progress beyond the state of the art by developing data discovery tools and tools which can be used to reflect upon and evaluate the quality of experiential rather than digital data.

In relation to Data Management, there are several international initiatives, starting with the FP4-ESPRIT 4 project (?, ?), that define a model in terms of data warehouse quality (?, ?) and more recently the DAMA International working on the Data Management Framework (?, ?). The last, describes the DAMA-PMBOK2 Framework which collects eleven (11) knowledge areas that are considered as best practices within the Data

Management discipline and seven (7) environmental elements that explain each knowledge area.

The knowledge areas represented by DAMA-PMBOK2 are: Data Governance, Data Architecture, Data Modeling and Design, Data Storage and Operations, Data Security, Data Integration and Interoperability, Documents and Content, Reference and Master Data, Data Warehousing and Business Intelligence, Metadata and Data Quality. In terms of environmental elements, enumerates the following: Goals and Principles, Activities, Deliverables, Roles and Responsibilities, Practices and Techniques, Tools and Organization and Culture, all of them, grouped by descriptor type (People, Process or Technology).

Regarding crime science, TRE$_S$PASS adopts two notable frameworks in this context are: crime prevention through environmental design, abbreviated CPTED (?, ?), and the 25 techniques for crime prevention proposed by the Center for Problem-Oriented Policing (?, ?).

In terms of levels of measurements, TRE$_S$PASS aims at quantitative risk management, in the sense that countermeasures can be selected based on an estimation of the return on security investment, the data management process needs therefore, more precise definitions of what are often called levels or scales of measurement (?, ?).

1.5. Approach

The approach which was followed to build this deliverable is, in short, enumerated by the following items:

- Gather deliverables that have worked on the development in the data management process.
- Analyse different topics of the data management process (data extraction, social and technical data consolidation, interaction with external actors.
- For each topic, revise old requirements and derive new ones.

More specifically, we gather the knowledge from the following list of TRE$_S$PASS project deliverables:

- D1.3.1 - initial prototype of the model (?, ?)
- D2.1.1 - initial requirements for data handling (?, ?)
- D2.2.1 - handling technical data (?, ?)
- D3.3.1 - stochastic analysis methods (?, ?)
- D5.3.1 - abstraction levels for model sharing (?, ?)
- D6.2.1 - requirements (?, ?)
- D7.1.1 - first deliverable describing the three case studies (?, ?)

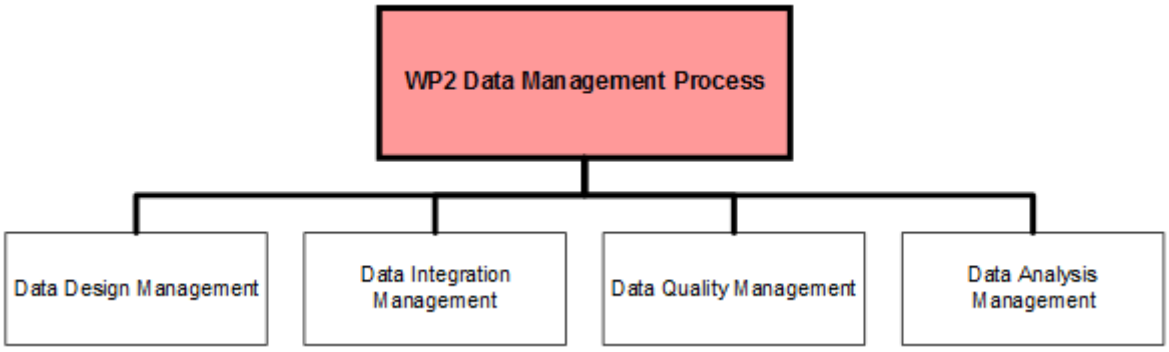

Figure 1.3.: WP2 Data Management Requirements Organization

Compared to the previous deliverable, this deliverable gathers experiences from preliminary analyses of all three project case studies. Furthermore, the deliverable builds upon the practical implementation and integration experiences from the ongoing prototype and demo development, as presented at the M12 review.

The requirements from the data management process were organized into logical components, as described below. The main reason is to facilitate the bridge between the project and the industry. The structure was designed to be simple and straightforward, and includes: design (concepts), integration (implementation), quality (validation), analysis (added value).

Finally, the following knowledge areas regarding data management are identified for the components, which guide the consolidation of the data requirements:

- Design: domains, concepts and semantics; types and properties; modeling, relations and constraints; control metadata
- Integration: source discovery; connection and transmission; representation and transformation; storage and persistence; privacy, access and protection
- Quality: cleaning and integrity; quality attributes and measures; maintenance and versioning; unknown, undefined and estimation
- Analysis: filtering; aggregations; pattern matching

2. Data management process

This chapter describes the requirements that refer to the development of the data extraction phase. Describes design and architectural requirements for the data management process.

2.1. Project progress on the topic of data extraction

Currently, the topic of data extraction is already present, as it has been introduced previously. In the case of the IPTV case study, prototyping tools started to be developed which extract data from the NMAP tool outputs producing a WP1 compliant input. Also the process of data extraction was a topic for the previous requirements deliverable on data management process, which identifies user communities of data, data discovery key issues, data retrieval (both automatic for technical data and manual for social data), metadata elements definition, and data storage. However, some of these topics not only need to be developed further and new ones must be introduced in order to fully support new project discoveries and the work produced within the case studies.

2.2. Revision of the initial requirements

Based on the obtained knowledge during project advances, we performed the refinement of initial requirements regarding the data extraction, first introduced in D2.1.1 (?, ?). In Section 2.2.1 we outline the baseline of initial requirements and in Section 2.2.2 we discuss the validity of the requirements.

2.2.1. General baseline of data requirements

In the document D2.1.1 (?, ?) we identify a set of initial requirements for the data management process. The following requirements relate to the data management and extraction:

- **IR2.1:** The methods used to extract data must be suitable for the data extraction goal: deliver data to the WP3 analysis in the WP2 model

- **IR2.2:** Main objectives of the data management processes are: the design of the data flows and internal manipulation (including types, relations, metadata); the integration with other stakeholders (including discovery, collection, transformation, delivery); quality control (including integrity checks, handling with uncertainty); analysis (including filtering, data inference)
- **IR2.3:** Data discovery and extraction process must be designed in a way that enables local adaptations and configurations. Varieties of environments in which the TRE$_S$PASS tools might be deployed, have different available data sources and different kinds of data. Therefore we must not rely on or assume the existence nor availability of any particular data source, but rather design the data discovery and extraction process to be sufficiently adaptive to work with the data sources available in the environment where the TRE$_S$PASS tools are deployed
- **IR2.4:** The methods used to extract data must be suitable for the data extraction goal: deliver data to the WP3 analysis according to the WP1 model.
- **IR2.5:** All data extraction must be conducted under the appropriate terms and conditions (from D2.1.1, Section 6.2) (?, ?)
- **IR2.6:** The extracted data must be defined in terms of metadata (Section 2.3.2.1) characteristics and appropriate format

This can be summarize into a single principle, which will be used to develop WP2 requirements, together with the other WP's requirements:

Principle WP2.1: All data extraction must be conducted under the appropriate terms and conditions and be defined in terms of metadata characteristics

From this principle, a number of conclusions can be inferred, including:

- Main objective of the data management processes are: the design of the data flows and internal manipulation (including types, relations, metadata); the integration with other stakeholders (including discovery, collection, transformation, delivery); quality control (including integrity checks, handling with uncertainty); analysis (including filtering, data inference)
- Data discovery and extraction process must be designed in a way that enables local adaptations and configurations. Varieties of environments in which the TRE$_S$PASS tools might be deployed, have different available data sources and different kinds of data. Therefore we must not rely on or assume the existence nor availability of any particular data source, but rather design the data discovery and extraction process to be sufficiently adaptive to work with the data sources available in the environment where the TRE$_S$PASS tools are deployed.
- Where appropriate for the data type, the data extraction must be automated or as automated as possible. This might include social data, although automated extraction might not be possible, for example when extracting data from non-digital means or non-structured data (free text) requiring manual or semi-manual processes. All scenarios where data is in a digital format and contain the minimum information needed by TRE$_S$PASS, automated processes must be prepared

- Methods for reliable data collection for likelihoods, costs and assets values must be identified. If not enough data is present, there must be a way to derive such parameters like costs, likelihoods of success, and assets values based on the available data and fall back to experts estimations if no data is available

For the data discovery requirement, an example in case of the Telco case study we know that the Call Records' Database is available as the data source, thus data extraction should be capable of extracting and analysing data from this source.

In this document we perform a refinement of the initial requirements based on research discoveries and the obtained knowledge on case studies

2.2.2. Refinement of general requirements

Most important aspect to recall regarding the extraction process is the challenge to extract unstructured or poorly structured data from data sources, especially into the social domain. In addition, intermediate extracting processes might be required to store data before the consolidation process, followed by data transformation and addition of metadata prior to proceed to the next stage (consolidation).

Adding structure to unstructured data might rely on a number of techniques, depending on the goals and the data itself. In the end, the objective is to have the data into a database model (entity-relationship).

- Standard sections (fixed) Identification, like for example particular fields
- Text pattern matching such as regular expressions to identify data structures
- Text analytic to attempt to understand the text and link it to other information
- Manual extraction of the data into the database system

2.3. Requirements for the extraction prototype

In this section we discuss intermediate requirements of the data extraction prototype. We structure our discussion along three main aspects: (a) required data sources, (b) required data types and (c) common design patterns. For (a) and (b) we build upon the earlier sets of requirements, while for (c) we derive a new set of requirements.

2.3.1. Required data sources

In this section we revise the set of data sources, initially identified in D2.1.1 (?, ?) and D2.2.1 (?, ?).

In the previous deliverable, a number of needs and conclusions were raised, including the need to identify user communities to support the data extraction process, and a number of possible data sources to extract data. While technical data must be retrieved automatically, social data might rely on manual procedures. Finally, storing data requires not only a data representation but also defining metadata elements to support the data.

2.3.1.1. Data sources

D2.2.1 (?, ?) outlines that there must be a distinction between *infrastructure data*, *attacker data*, and *real-time data on actions* (which would allow to derive quantitative properties of infrastructure and attackers). The following list of initial data sources is outlined:

- Quantitative penetration testing: the relevant data, such as time and effort is recorded and may be used to derive difficulty (strength of the system component) separately from the attacker skill level
- IDS/IPS systems
- Security and event logs
- Perimeter security elements (i.e. firewalls conducting deep packet inspection)
- Social science experiments in social engineering
- Social science inductive research. These may establish patterns for data protection practice.
- Historical data from EU institutions (i.e. Europol)
- Information about known security incidents.
- CCTV surveillance systems.

2.3.1.2. Revised data sources

A major concept to connect and extract data from data sources are adaptors, which are responsible to handle data connection issues including different message transport protocols and message format protocols. In terms of the message format protocols, the ability to convert one data format, specific to a certain data source, into a TRE_SPASS readable format rely on those adaptors. Data transformation logic required to perform, might be complex and in some cases not fully (only partially) possible.

A plug-in oriented approach is recommended to implement the process of data extraction from the data sources, where for each individual type of data source, a plugin is developed

and then plugged into the overall architecture, creating a layer of isolation between the data access and the data. This should be applied to both connection/transport protocols and message format, where a layer of isolation between these two should exist in order to guarantee the independence between transport protocols (i.e. HTTP, FTP, NTFS, etc.) and message format protocols (i.e. TRESPASS-XML, SOAP, HTML, CSV, etc.).

2.3.2. Required data types

In this section we revise the set of data types, derived in D2.1.1 (?, ?), that were initially considered as relevant for the data management process in WP2.

Data types already identified within the project are listed below.

Table 2.1.: Summary of required data

Model Type	**Type of data**	**Required data and mesurements**
Socio-technical	Potential Targets	List of classified targets
Socio-technical	Potential Targets	List of vulnerabilities, by target
Socio-technical	Potential Targets	Likelihood of attacks success by target
Socio-technical	Potential Targets	Impact of attack, by target
Socio-technical	Potential Targets	Statistical data on incident reports by target
Socio-technical	Potential Targets	Security policies by target
Socio-technical	Potential Targets	Value by target
Socio-technical	Technical Infrastructure/Topology	Technical infrastructure
Socio-technical	Technical Infrastructure/Topology	Technology assets/asset groups
Socio-technical	Technical Infrastructure/Topology	Asset-level security policies
Socio-technical	Technical Infrastructure/Topology	Network topology
Socio-technical	Technical Infrastructure/Topology	Geolocation of assets
Socio-technical	Technical Infrastructure/Topology	List of known vulnerabilities per asset
Socio-technical	Technical Infrastructure/Topology	Organisation's incidents logs
Socio-technical	Governance architecture	Organisational hierarchy
Socio-technical	Governance architecture	Internal control culture
Socio-technical	Governance architecture	Asset/network/infrastructure responsibility and ownership
Socio-technical	Governance architecture	Business processes and goals
Socio-technical	Governance architecture	Organisational security policies
Socio-technical	Governance architecture	Legal/Compliance issues

Socio-technical	User Community Data	Data handling practices
Socio-technical	User Community Data	Values and data handling norms
Socio-technical	User Community Data	Compliance performance
Socio-technical	Information Storage and Flows	Identification of information flows
Socio-technical	Information Storage and Flows	Identification of stored data
Socio-technical	Information Storage and Flows	Identification of classified data
Attacker	Attack step	Difficulty
Attacker	Attack step	Resources spent
Attacker	Attack step	Likelihood of success
Attacker	Attack step	Impact
Attacker	Attack step	Statistical data on incident reports
Attacker	Attack step	Value
Attacker	Attack step	Likelihood of detection
Attacker	Attacker	Nature of attacker
Attacker	Attacker	Goal
Attacker	Attacker	Behaviour
Attacker	Attacker	Possible patterns of attack
Attacker	Attacker	Motivation
Attacker	Attacker	Capabilities
Attacker	Attacker	Resources
Attacker	Attacker	Strategy
Attacker	Attacker	Knowledge about organisation

2.3.2.1. Control metadata

Previously it was stated that data from multiple sources have metadata characteristics that assist with interpretation. The following elements where introduced: security classification, durability, collection dates, owner, provider, reliability, and so on.

Going into more detail data might come from several different data sources to be consolidated into the TRE$_S$PASS concepts and format. In order to do so, the data management process must be able to support a number of functionalities (in addition to the usual regarding data modelling, like data type), including for example traceability, data versioning, consistency control, etc. In order to support these data functionalities control metadata applied at the data record level might be needed to include within the data itself, including the following:

- Status: (active, inactive, removed, etc.) data should avoid being physically removed, but updated to a removed status

- Creation and last updated owner
- Creation and last updated timestamps
- Version token: must allow ordering
- Checksum token: hash

These metadata attributes must be enough to guarantee or identify a number of data quality properties including (?, ?, ?):

- Accuracy: proximity of the data to the true value (which may be used to measure the data confidence level)
- Correctness: correct with respect to a specification, or in this case a data model (which may be used to measure the data error level)
- Completeness: without gaps or missing data (which may be used to measure the level of missing data and subsequently the level of unknown)
- Relevance: how well the data meets the information needs (which may be used to measure the level of utility of the data)

A data quality process should analyse the above data properties and then classify it in accordance.

2.3.3. Representing uncertainty

Under the topic related to the data levels of measurement, TRE$_S$PASS aims to perform quantitative risk management, in the sense that countermeasures can be selected based on an estimation of the return on security investment. Therefore, the data management process needs precise definitions of what are often called levels or scales of measurement. The following concepts where described previously: interpretative data, may involve elements which cannot easily be simplified to the level of a code like social and cultural factors. In terms of codified data, which express a value in a certain domain, a common distinction between nominal, ordinal, interval and ratio scales exist.

As not all data can be measured precisely and representing uncertainty in the data must be possible. As a result, ordinal scales such as low, medium, high; might be used or, in alternative to use intervals to represent data. Another option will be to use probability distributions for the variables, to express the likelihood of the value being within certain limits. This is the most expressive form, but it requires some level of calculation, and representation since we might need to represent the probability function and not only the values. An approach is presented in the subsequent paragraphs.

To represent both discrete and continuous distributions, functions may also be used (i.e. normal distribution, exponential distribution, etc.) instead of pre calculated values (which only apply on discrete distributions). The implementation of those functions must follow a naming convention and interface access rule (i.e. double <function name>(double V, double M, double S)) in order to allow a dynamic binding call mechanism during runtime,

to calculate the result giving the parameters set of values (i.e. value, mean, standard deviation) The idea to have more than one way to represent the same concept (i.e. a probability result by its value or, in alternative, by its function and the parameters set) is to be as flexible as possible and to guarantee adaptability since different data sources and data targets might contain, require or be compatible only with one of the formats. And the reason not to use only one form (i.e. always use discrete distributions pre calculated values, which in this case can be seen as the fits all solution), applying a transformation when another form is required, is because usually during this kind of conversions (i.e. between a continuous distribution to a corresponding discrete distribution), some level of information accuracy is lost that should be delivered when the data target supports it.

Principle WP2.2: Data types must be precisely defined as much as possible (strong types), and domain and interval values (data ranges) data types, must be pre-defined as much as possible.

From these requirements, a number of conclusions can be inferred, including:

- Data attributes must be strong typed as much as possible (for example avoid using a free text data type as a way to represent data) in order to be better prepared for automatic processing
- Domain values (well known list of possible values) must be used whenever possible to facilitate data processing
- Interval values (i.e. confidence intervals, credible intervals, etc.) may also be represented using minimum/maximum value or through domain values representing well known values
- Probabilities distribution used to represent uncertainty, should be represented through pre calculated values, like discrete probability distribution (a sequence of the distribution pre calculated values), including continuous probability distributions

2.3.4. Attacker behavioural patterns

A driver for all the data management requirements is the analysis of the various roles an attacker in deliberately perpetrating unauthorized actions which jeopardize the target system.

It is then important to study the attacker profile and define a TRE$_S$PASS actor which gathers all forms possible. The attacker can be seen as an abstract entity, which may take the form of a person, organisation, IT equipment, piece of software, or any other. It can be external or internal to the target organisation, and range from pranksters, disgruntled employees, to organised criminals, terrorists and foreign hostile governmental institutions. When entering the military domain of cyber warfare, significant resources and legal protection is at the disposable of the attacker.

While the motivations and modus operandi of attackers are mostly known, and behavioural patterns are widely studied, both technology evolution and motivations (financial, political,

etc.) will always lead to new different threats and maintain non-negligible risk levels to information security.

The TRESPASS attacker actor will consider active types of attackers, (i.e. those which are known to have attacks records or pose imminent danger); and not active ones. The ones not active at the time, exist but take a neutral or even friendly stand, which can abruptly take a turn shall motivations change or vulnerabilities appear. The regulator and the authorities can assume the role of an attacker, as in the case of Homeland Security departments, and information agencies running unethical activities, acting according to the law or against it. In the specific case of equipment or software being threat, one must consider the case were friendly security purposed products are unreliable due to ownership. Some mass market security products are known to serve the interests of institutions behind the seller shareholding, when suitable. Actors are hard to monitor regarding cloud computing services, when offered by large companies listed in international stock markets, whose real owners are unknown, masked by layers of companies all over the world.

The key properties defining the TRESPASS attacker actor are the:

- Classification
  - Type: human, equipment, software, unknown
  - Dimension: individual, organisation, unknown
  - Location: internal, external, unknown
  - Hostility: friendly, neutral, hostile, unknown
  - Legal form: citizen, criminal organisation, national institutional, external institutional, unknown
  - Experiece: professional, expert, amateur, opportunistic, unknown
- Risk to target organisation
  - Level: high, medium, low, none

2.3.5. Data design patterns

As the main objective, WP2 strives to produce a software tool that can perform automated extraction and processing of data. To address this objective, a software prototype is being developed. In this section we derive more specific requirements on the design of the software. We base our analysis on the set of common design patterns that are adjusted to the TRESPASS and discussed in the context of identified challenges from the initial extraction prototype. In Section 2.3.5.1 we present common software design patterns. In Section 2.3.5.2 we present new design requirements, in the context of TRESPASS data extraction prototype.

2.3.5.1. Common data design patterns

There are many aspects to consider in the design of a piece of software. The importance of each should reflect the goals the software is trying to achieve. Some of these aspects are:

- Compatibility - The software is able to operate with other products that are designed for interoperability with another product. For example, a piece of software may be backward-compatible with an older version of itself
- Extensibility - New capabilities can be added to the software without major changes to the underlying architecture
- Fault-tolerance - The software is resistant to and able to recover from component failure
- Maintainability - The software can be restored to a specified condition within a specified period of time. For example, anti-virus software may include the ability to periodically receive virus definition updates in order to maintain the software's effectiveness
- Modularity - the resulting software comprises well defined, independent components. That leads to better maintainability. The components could be then implemented and tested in isolation before being integrated to form a desired software system. This allows division of work in a software development project
- Reliability - The software is able to perform a required function under stated conditions for a specified period of time
- Re-usability - the software is able to add further features and modification with slight or no modification
- Robustness - The software is able to operate under stress or tolerate unpredictable or invalid input. For example, it can be designed with a resilience to low memory conditions
- Security - The software is able to withstand hostile acts and influences
- Usability - The software user interface must be usable for its target user/audience. Default values for the parameters must be chosen so that they are a good choice for the majority of the users

2.3.5.2. Design requirements for the data extraction prototype

In this initial stage, the key aspects that must be considered in the TRE$_S$PASS extraction prototype are related to the flexibility of supporting several types of data and formats and to transform them into a TRE$_S$PASS format. Those can be enumerated by: extensibility, modularity and re-usability.

2.3.6. Dealing with extensions - a practical example

Recalling the ATM case study, a need to extend the model with additional characterization is present, more precisely the geographic nature of the case study. This case study includes two levels of analysis. The first level is a statistical level of the territory in terms of propensity for attacks (i.e. some area of the territory was higher levels of propensity than others: crime occurences are not consistents over the territory). In order to infer this first level, socio demographic, socio economic, environmental, historical events, data is used. This statistical level provides a high level overview of this kind of attacks, and its very much useful to the authorities (the police), because provide information on how optimize defensive resources over the territory at a higher level (where to reinforce patrols, etc.). Then, a second level is a TRE$_S$PASS attack-tree analysis, that over the statistical level, provides a detailed level of information that takes into consideration the very specific context where the machines are deployed (building, doors, walls, cameras, technical detail of the machines, etc.). This detailed level provides a specific detail of this kind of attacks, and its very much useful to the infrastructure owners (the bank), because provide information on how to optimize defensive resources in the specific ATM locations. Therefore, both levels complement each other seice the detailed level must take into consideration the statistic level evaluation: two machines that are exactly the same, deployed in two buildings that are exactly the same like a gas station with the same building setup (detailed level), can have completely different risk evaluations (likelihood of being attacked), because one is deployed in a social problematic zone (statistic level)while the other is not. Both are oriented to different end users: from police authorities (statistic level), to infrastructure owers (detailed level). For that propose, we'll start by extending the TRE$_S$PASS base model with geographic features, since the statistic level relys on it. Other extensions to the base model are, of course, possible. The most important message is that, in order to allow different scenarios of application, the TRE$_S$PASS base model must be flexible enough to accommodate extension to the data model and processes. Crime analysis involves the collection and analysis of data related to a criminal incident (attack), offender (attacker), and target (victim). In this context, a TRE$_S$PASS data management process must understand and be aware of relevant aspects related to crime analysis data, more precisely being able to support the identification and generation of the information needed to assist the analysis and decision processes and the deployment of efficient countermeasures to prevent and reduce the likelihood level of criminal activity. Data is a critical part of the crime analysis and the following properties must be present (?, ?):

- Relevant: include relevant elements to perform crime analysis (incident data on the type of crime, location, time, date, target, suspect, property stolen or affected, modus operandi)
- Reliable: multiple data sources and procedures must be set up to ensure the continued collection and processing of data
- Accurate: data must be reviewed and cross checked against different sources

- Timely: if not delivered on a reasonable period of time, successful attacks might be performed in the meanwhile
- Comparability: data must be able to compare over time, region or other domains

Now, developing deeper the relevant property of the data, we can distinguish now between physical crime and cybercrime. Physical crime has an inherently geographic quality and is not randomly distributed. Criminal occurrences happen where offender awareness space and opportunities coincide in the same moment of time. The question now is: does the same principle apply to crime in cyberspace, even though the criminal interaction is not a physical one? Cyberspace is an amorphous space that does not occupy a physical or geographical location. Cyberspace has no geographical boundaries: it establishes immediate long-distance communications with anyone who can have access to any internet endpoint, and can take place across various jurisdictions and hence the legal issue of jurisdiction. Usually an internet user has no way of knowing exactly where the information on a site is being accessed from. Nevertheless, since TRESPASS aims to create an integrated view and analysis of crime data from different sources (technical, social, physical).

Examples of possible applications are physical infrastructures or equipment operated remotely by computer software programs exposed in the internet which are increasingly common (i.e. water pumps, rail-road switches, ATM's, IPTV equipment's, etc.), that can be hacked by an attacker that manages to access and violate their software control programs through the cyberspace. In these cases, the geographic context of the attacker might be irrelevant to understand the attack since the attacker can be accessing the cyberspace from elsewhere; however the victims of the attack might be selected because of a number of items, including the geographic location (specially operating equipments, more than application servers)

The location concept can be seen as composition of two vectors that define the security characteristics of a location. By one side you have the location type that defines the social and technical characteristics of a location. An example of a location type is an office. Although the concept of office is easily understandable and defined in any location in world the security associated with an office in Lisbon or in Luanda will be different due to social characteristics of each city. Hence geographical location will constitute the second vector defining the security characteristics of a location. Locations can be connected by edges that define the rules and effort required for moving from one location to the other. Edges are unidirectional since moving into or out of an office will have different rules. An edge is composed of a set barriers that have certain rules and effort required to be crossed. For instance a barrier from moving from the street to the office might be the turnstyle with keycard at the reception of the office. By defining conceptual barriers with rules and effort for the barrier to be crossed, the security risks can analysed in a systematic way. Depending on the domain the effort to move between geographical locations can be different. In a real world domain the effort for an actor to move between geographical locations is proportional to geographical distance but the internet domain the effort to "move" an artefact from one location to the other is nearly zero if other barriers are not imposed.

The attacker goal can be conceptually defined by a set of locations or sequence of locations. The goal will be reached if an actor or artefact can reach a location or move across a sequence of locations. Depending on the domain the goal can be to reach a location where the attack can be perpetrated or "moving" an artefact to a location where the attack can be perpetrated. Either way the goal will be depending on a location.

From this perspective, and in order to model crime data (attack, attacker, victim) to perform crime analysis, three data types will be considered: socio demographic, temporal and spatial.

- Socio demographic: personal characteristics (i.e. sex, race, income, age, education, etc.) of individuals (to identify crime suspects and victims), and groups (why people from one group are more victimized than people from other group, etc.). Used to classify attackers and victims
- Temporal: levels of temporal analysis, which include the examination of crime trends over a long period of time (i.e. season, day of week, time of day, etc.). Used to classify attacks
- Spatial: the spatial nature of the crime (geographic location), which focus on understanding geographic patterns of crime by examining situations in which victims and attackers come together in time and space. Used to classify attacks

Spatial data can be represented using geographical entities (points, lines, polygons), located using a spatial (or coordinate) reference system. To capture, store, manipulate, analyse, manage, and present all types of geographical data, a geographic information system (GIS) must be used.

The technique used to aggregate all this characterization data types is through pattern detection (data mining). Using pattern detection might conduct to interesting results in crime analysis, which must include the analysis of attack, attacker, victim correlations, including attacker and victim profiling, and attacker behaviour patterns. Pattern detection occurs when attacks reported during a certain (preferable short) period of time have common attributes, which can be one or several of the characterization data types identified previously. Several patterns might also exist each one with a different level of relevance (likelihood of occurrence). Attention should be taken to the fact that each individual characterization data type might also have a difference level of relevance (or weight) in the pattern. For example age and education might be part of the pattern but age shows to be far more relevant than education (i.e. 75-25 percentage), when evaluating the pattern.

Giving an example, let us consider an ATM attack pattern, consisting on performing a fraud where the attacker somehow hacks the ATM operating system programming it in a way that he manage to withdraw all the money that is inside the ATM. The following pattern can be considered as credible (high level of relevance):

- Socio demographic: sex: male; education: technical education; age: between 25 and 40 years old
- Temporal: during the night between 02h00 and 04h00

- Spatial: rural environment: isolated ATM (no residential houses within a 200m radius), low populated area, no public illumination nearby

Social demographic data can be used to characterize a TRESPASS attacker. Temporal and spatial data can be used to calculate the attack likelihood.

In the provided example, identifying the spatial pattern described (isolated ATM, low populated area, no public illumination nearby), allows us to project the same spatial pattern over a larger area (find ATM's on a similar spatial environment), in order to identify ATM's that according to the pattern are predicted to have a high level of vulnerability, and consequently should be taken into consideration during the countermeasures priority analysis of the prevention process. Same logic applies to other data types rather than spatial.

However, an important aspect of spatial data is that manipulating spatial data is different than manipulating alphanumeric data (i.e. numeric, logical, textual, etc.), since the operations defined for the spatial data types (geometries: i.e. intersects, contains, distance between, etc.) are not the same used to manipulate numeric data types (numbers: equals, higher, lower, etc.), logical data types (boolean: and, or, not, etc.), and others. Another issue is that detecting spatial patterns (spatial data mining) in this context is complex since it is more than searching for events concentration (i.e. attacks performed over a certain location or spatial boundary), but to find correlations between different spatial data types (i.e. low populated area without public illumination nearby). In the limit, infinite spatial correlation elements (i.e. roads, houses, street lights, etc.) might exist. A technical short-cut to overcome this might be to allow the introduction of a spatial pattern provided by expert judgement where the spatial correlation elements are defined in advance.

Finally, attack patterns must be updated over time in order to adapt to the new attack trends. This way, as the environment (including spatial patterns) and attacker behaviour changes over time, these patterns follow those changes.

In conclusion, understanding concepts related to crime analysis and the types of data used to characterize those, can be used to generate patterns related to crime activity. Those types of data can be divided into: socio demographic, temporal and spatial; and data mining techniques can be used to analyse and identify crime patterns. These patterns can be very useful for the risk analysis process and countermeasures prioritization, since it provides hints about attacker and attack patterns. Spatial data introduces new challenges to the data mining (spatial data mining), associated with the different nature of the spatial information; however it can be very useful if not critical when the physical domain is present. The main limitation is that in order to generate these patterns, a crime analysis process of data collection must exist, which relies on the characterization of past attacks, which means that this method is not suitable to be used in identifying new forms of crimes. Nevertheless it still covers a relevant number of crime scenarios.

Principle WP2.3: TRESPASS data must be segregated at domains (technical, social, physical, etc.) and updated regularly.

From this requirements, a number of conclusions can be inferred, including:

- In order to model data (attack, attacker, victim), to perform a complete risk analysis, at least two data domains should to be considered: technical and social. Additional domains might include: physical, business model, and others. In addition, data might be complemented using: temporal information, location and other means of classification
- Data mining as part of data analysis processes, or other type of data inference ability to identify attributes correlation between attacks, attacker, victim, must be considered as a way to add value to the data
- Expert judgement can be considered to complement and fill gapes in the existing data, and also to allow prediction. This expert judgement include data from all the data domains
- Attack patterns must be updated over time in order to adapt the model to the new attack trends, supporting possible prediction methods

2.3.7. Creating a TRE$_S$PASS data model ready to be extended

The TRE$_S$PASS model must represent attributes including numeric values like cost, likelihood values and probability distributions, difficulty, time, skill, etc., to be processed by other WP's or requested directly from the WP's to WP2. Therefore, WP2 must understand TRE$_S$PASS base concepts, find data sources or alternatives. Each concept and attribute must be validated to understand which ones are generic to any TRE$_S$PASS implementation and which ones are specific to a certain context (i.e. IPTV, Cloud, Telco, etc., industry/market). WP2 must define this boundary between generic and specific concepts and attributes, and take that into consideration while building a data model.

In addition, other TRE$_S$PASS concepts must be represented, in a way that every individual TRE$_S$PASS implementation can develop their particular knowledge base/experience (attacker libraries) over time (specific attack definitions, attacker profiles, etc.), that can be exchanged with other TRE$_S$PASS related domains implementation:

- Attack definitions (attack trees, fault trees, etc.), including structures (trees, graphs, etc.)
- Attacker goals and attacker profiles (linked with social behaviours) (D2.1.1 (?, ?), D2.1.2)
- Social vulnerabilities (D2.3.1 (?, ?))
- Cost/benefit analysis including countermeasures library
- Unknown (likelihood, level of reliability, confidence, overall impact on the calculated values, etc.)

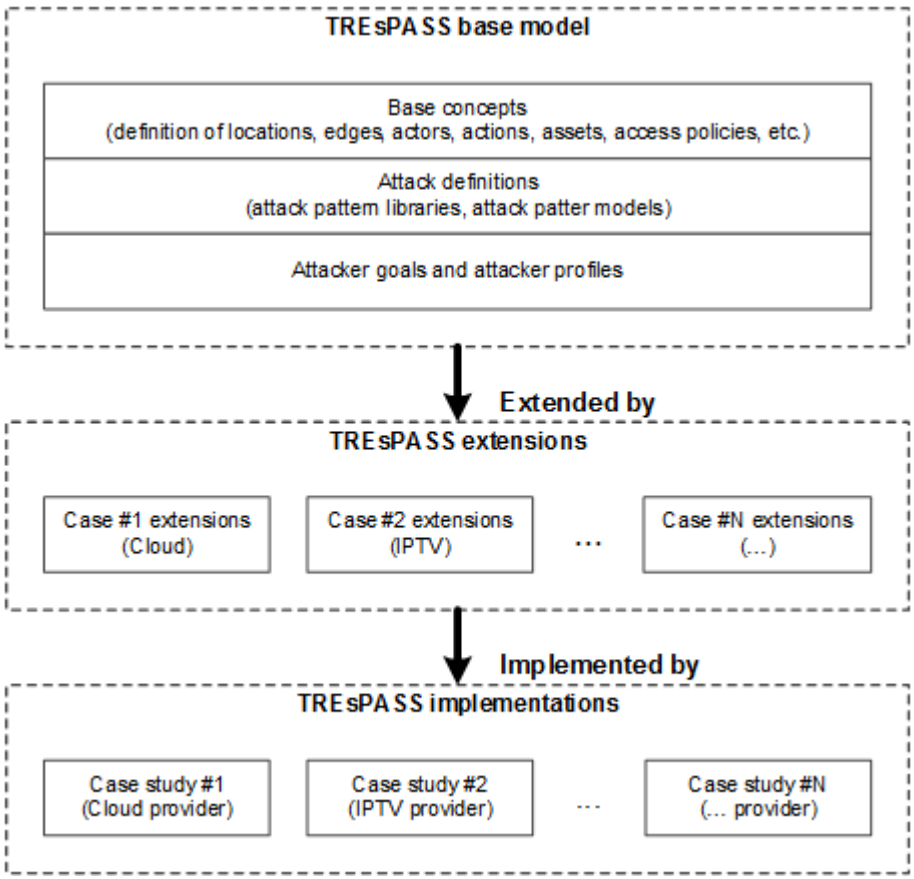

Figure 2.1.: Data hierarchy model from TRE$_S$PASS base concepts to specific TRE$_S$PASS implementations

Principle WP2.4: Attributes including numeric values like cost, likelihood values and probabilities, difficulty, time, skill, must be supported. Additional attributes can be inherited from the TRE$_S$PASS base model

From this requirements, a number of conclusions can be inferred, including:

- Attributes including numeric values like cost, likelihood values and probability distributions, difficulty, time, skill, etc., must be supported
- Each attribute must be inherited from the generic TRE$_S$PASS model, otherwise will be specific to a particular context (an extension to the model).

The TRE$_S$PASS model will include the common attributes for each entity. For each specific domain additional attributes and methods can used to extend the TRE$_S$PASS entities. These additional attributes and methods should only be used to model characteristics that are only relevant within the domain(s) for which they are defined.

The generic TRE$_S$PASS tools will work seamlessly across all domains since these tools will use only the generic attributes or methods. This doesn't invalidate that methods are implemented differently in each domain and depending on the domain characteristics. The

TRESPASS model entities will act as classes that can be extended and have methods refined within each domain.

This object-oriented approach will allow to have a strong common model without losing the flexibility needed to model each specific domain. The core model will focus on the security model while extensions will cover specific aspects of each domain. Generic methods will be inherited by the domains but in each domain a specific method can be redefined if the generic implementation doesn't cover the specificities of the domain.

2.3.8. Cost of the data and cost benefit analysis

Taking the Telco case study as example, from the final user perspective, the initial consideration of adopting a technology like TRESPASS will not be the analysis process but to have a perspective about the overall advantage of adopting the technology. The initial step will be to understand if the money value the institution will save compensates the cost of adopting the technology. In the case of the Telco case study, we can easily fall in a situation where even if the final user agrees that applying TRESPASS will add a value (review the tariff misuse for call termination scenario), the decision might be that, as long as the losses do not exceed a certain threshold limit the company will simply ignore this kind of fraud.

This is different from the countermeasures cost benefit analysis as part of the analysis process, because it's a preliminary step. The reason is that, adopting any technology requires resources, including funding, and in order to proceed (or initiate), there must be some idea of the final advantages. This can be done by the means of a business case, and the first steps to build it are collecting data.

A cost benefit analysis must be made in order to plan, implement and operate data extraction mechanisms. The analysis makes sense looking at the overall TRESPASS data management, as it would not be complete without the full exploitation of the data, i.e., considerations on implementing countermeasures and results expected. Since there is investment to be made, a business-like model approach is to be followed. The model must contain Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) variables, balanced against the Return on Investment (ROI) made. The CAPEX consists on expenses from equipment, personnel training, preparation security audits, SW licences (data extraction or not), consultancy and other procurement. The OPEX deals with licence renewals (data extraction or not), maintenance, manpower and other costs directly imputable to additional security countermeasures.

The ROI calculation is a pure risk analysis, estimated from the mitigation or elimination of a loss derived from a security attack incident, comparing the situations where countermeasures under analysis are in place or not. The following diagram depicts this methodology (?, ?):

While it is relatively simple to calculate the countermeasure costs, the potential loss per security risk/incident is far more complex. It is highly dependent on the case, and there are no general formulae. Typical factors to be considered:

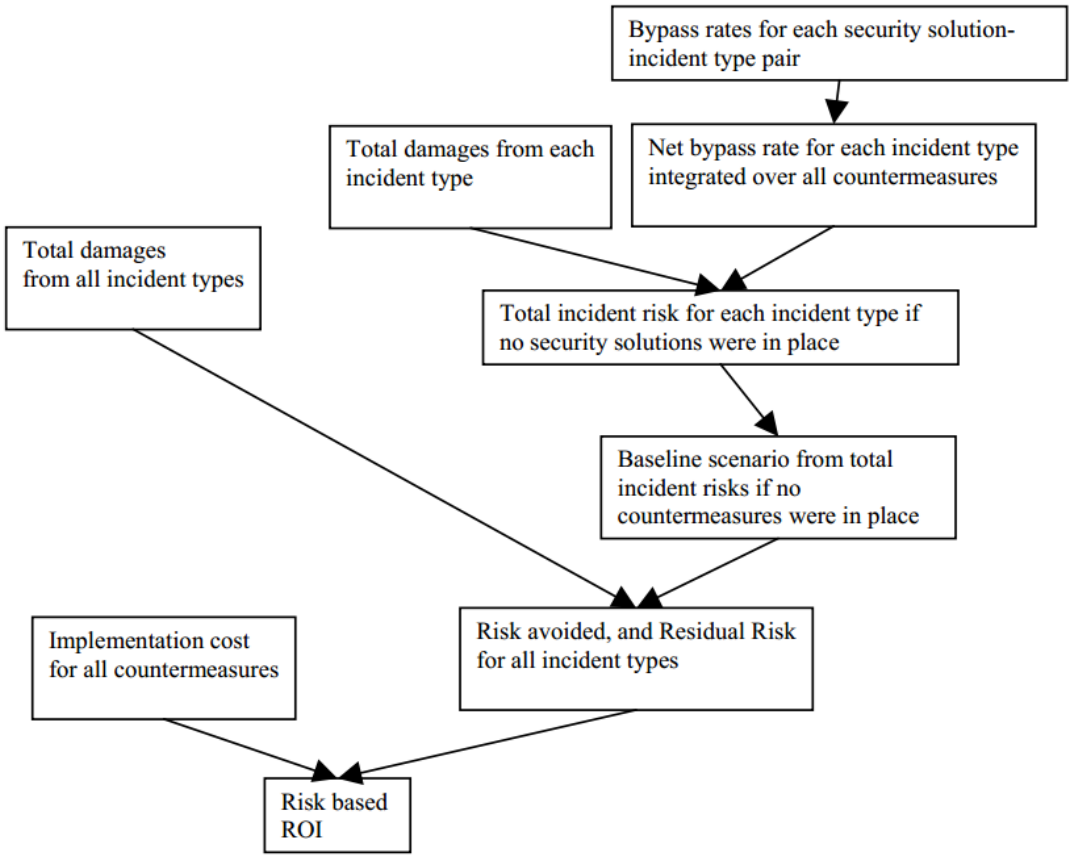

Figure 2.2.: Data ROI on data protection and prevention

- The cost to recover from attack, including productivity loss and effort from IT staff
- The cost to recover from a network attack, including reconfiguration locations and edges (firewalls, and other network assets) to prevent future incidents
- Legal penalties and revenue loss for violating confidentiality and privacy agreements due to security breach
- Revenue loss due to unavailable systems crucial to business operations, compromised by an attack

The cost-benefit analysis of investment on countermeasures must weigh the CAPEX+OPEX investment, the cost to recover of security attacks, and the likelihood of these happening.

Principle WP2.5: Data must allow the implementation of cost benefit analysis for the implementation of TRE$_S$PASS based security processses.

From this requirements, a number of conclusions can be inferred, including:

- Data extracted and transformed must allow the implementation of cost benefit analysis related to the application of countermeasures
- Being able to evaluate the investments needed to apply the countermeasures and the return of investment (ROI) analysis must also be possible
- Before implementing a TRE$_S$PASS framework into a real scenario, the framework must support the development of a business case about the gains of adopting this technology
- Data extraction process must also be able to consider the cost of data extraction and perform a cost benefit analysis about the data being collected and ignore data that does not met the cost benefit criteria (the cost of extracting the data does not compensate the advantage of having it)

3. Integration of social and technical data

This chapter describes possible approaches to handling both social and technical data in order to achieve the greatest utility from each. Where appropriate, requirements for integration of social and technical data are explored, in order to derive a more complete picture from the data available.

These requirements will depend upon the nature of the data available and also the processes defined by the users of that data.

In order to facilitate understanding of the consolidation process, the text is supported by a practical example, which will be an ATM case study. This case study was chosen because unlike other case studies, this one was not yet presented in detail on previous deliverables, and includes a number of needs for data consolidation from different domains (technical, social, environmental).

There are a number of major challenges in developing a data process to consolidate data from different domains which were collected differently, using different semantics, types, means and formats. Therefore, different case studies or applicable scenario, with different needs include different data management requirements, not only in terms of the data itself but also in the way input data is interpreted, transformed and processed by WP2 in order to derive output data according to the WP1 models, ready to be delivered to WP3.

Therefore, each specific model might contain new entities and attributes, additional metadata elements that expand the base model plus rules about consistency, transformation, data interpretation, processing and inference capabilities we'll call an extended model. The most important aspect to consider and understand if the fact that "model" here means not only a data model (like a entity/relationship data model), but also the rules and techniques applied to the data, which can be aggregated by case study or applicable scenario. For example the two already identified, in the previous deliverable, criminal models: Crime Prevention Through Environmental Design (CPTED) (?, ?); 25 Techniques for Crime Prevention (?, ?); include data requirements which include not only the kind of data needed, but also how data must be interpreted and prepared. When applying a given model, this must be taken into consideration, knowing that a rule applied to data from one model might not make sense to apply on data from the other model.

ATMs can be attacked through physical attacks, malware attacks, card skimming, and others. In these different scenarios, attackers behave completely differently, have completely different skills and ATM targets. In the limit, between an ATM physical attack where the machines are physically stolen to be broken, and the digital attacks where a software malware is installed to take control of the machine's operating system, these can be considered two different case studies, where ATM characteristics to perform a

vulnerability scan and a risk analysis are completely different (from the physical structure to the operating system) and distinct from one another.

Therefore, the project case studies have shown during their initial development that different scenarios might require different approaches. This is the reason why, the TRESPASS framework must be flexible enough to accommodate those by identifying generic concepts and by allowing the base model to be expanded.

Another reason is that, defining, representing and understanding different types of crime rely on different data concepts and attributes. To some types of crime, certain data types are critical and others irrelevant and vice versa (i.e. socio-demographic information about the attacker, attack physical location, victim profile, etc). An example is the attacker location that can be considered as irrelevant on the majority of cybercrime events, but critical to understand physical infrastructural attacks even if supported by a logical network attack.

The proposed method for data integragration is to define a conceptual baseline model and then to derive both the technical and social data from it, making consolidation possible since each domain specific data model has the same base (same data model and data rules). Without a common model the consolidation of data that represents different concepts and meanings will be impossible. The TRESPASS concepts will be introduced by the TRESPASS base model that must be in place and known by the data analysis processes.

Other challenges are related to data formats (i.e. social data can be gathered using surveys or other non-technical means). In those cases, a step to create technical data models to accommodate that data and the corresponding transformation operations will be required. This step might involve semi-automatic work and expert judgement.

3.1. Project progress on the topic of data consolidation

In general, the project is achieving progress in the data consolidation process. The current efforts mostly treat different data types separately, namely, T2.2 works on enabling automated extraction of technical data and the results are presented in D2.2.1 (?, ?), while T2.3 works on analysing different approaches for the extraction of social data and the results are presented in D2.3.1 (?, ?). In this deliverable we work on defining more specific requirements regarding the process of data analyses and integrity checking, storage and output. In this deliverable we introduce the data topic consolidation.

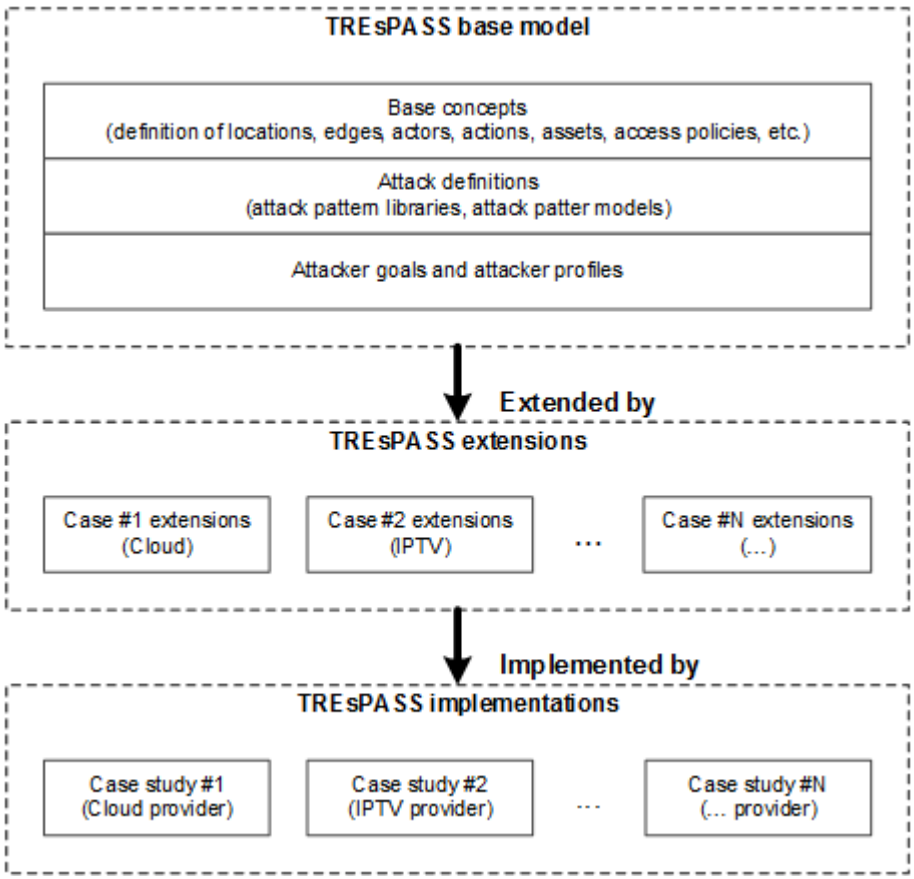

Figure 3.1.: Data hierarchy model

3.2. Understanding the integration process

3.2.1. A practical example - year 3 ATM case study

Different case studies and application scenarios will require different data management approaches. However we'll shortly introduce the ATM case study in terms of data management in order to facilitate the comprehension of the progress and how all these pieces can be placed together as a whole in a practical scenario.

ATMs are desirable devices for criminal attacks, because of the money contained in the safe. Therefore, different types of attacks to these machines have been recorded over the years. Classical attacks include physically stealing the machine in order to open it to access the safe (physical attack). Recent attacks include installing malware on the machine in order to take control of the ATM devices, including the ability to open the safe after a certain key code sequence is pressed through the keyboard (digital attacks). Other attacks are possible, but in this case study we'll focus on these two (physical and digital attacks). Some ATM attacks require physical contact with the machine, this mean that offender and victim (the ATM) must meet in time and space. Or, according to the

crime triangle theory, the criminal event occurs when a likely offender and suitable target come together in time and place, without a capable guardian present. Therefore not only technical aspects of the ATM infrastructure including the machines' place-holder matters (the technical part of the case study), but also the location of the machine matters, because of the potential attackers and behaviours (the social part of the case study). In this case study the objective is to develop a risk assessment of an ATM network taking into consideration the technological aspects of the network (technical data), and the social characterization (social data) of the places that surround the machines (where potential attackers move around). Environmental data can also be used: proximity to roads, existence of public illumination, buildings degradation status, etc. Therefore we'll need a data integration process to mix these different types of data (technical, social, environmental), and then some pre-processing, like correlating the data (for example: what socio demographic characteristics play the most relevant roles in ATM crime?), using historical data about attacks or experts judgement; in order to calculate a risk factor for each machine in the network, representing how likely can we expect a certain machine to be attacked. The case study output will be finalized by providing a cost benefit countermeasures recommendations for each machine and for the network as a whole.

In terms of data management process requirements are related to:

- Gather technical, social, environmental data from relevant existing data sources
- Consolidate and store the data from the different domains to allow a coherent data analysis
- Infer data from the existing data to add value to the process
- Validate and classify the data for consistency and completeness
- Transform the data into a format (WP1) ready to be delivered for fine tuned processed by the subsequent analysis processes (WP3)

In terms of the data itself requirements are related to:

- Technical domain (infrastructure)
- Social domain (socio-economic and socio-economic characterization)
- Environmental domain (other relevant data about the territory)
- Experts judgement to fill uncertainty gaps
- Historical data about attacks to validate the model
- Countermeasures data to allow a cost benefit analysis and recommendation

Specific to this case study is the geographic nature of the data, which will require extending the TRE_SPASS model with geographic capabilities. Nevertheless, this geographic data processing should be confined to the WP2 scope.

3.2.2. Data definitions

As said before, WP2 identifies, discovers, and categorises behavioural and infrastructure data, from both organisation-specific and more global sources, consolidating these social, technical and physical domains into a unified and consistent model of information (?, ?). WP2 needs to know which types of data are available in deployment environments and must be able to obtain the necessary data directly from there. As the basis, the TRE$_S$PASS model represents the following concepts (?, ?): access policy, location, actor, behaviour data, social vulnerabilities. In addition, there are a number of data classification items that must be taken into account when defining TRE$_S$PASS data management process, including modelling, representing and sharing. Those include: data levels of measurement, attacker data and attacker modelling and metadata elements.

The following subsections detail the requirements for both cases, technical and social.

3.2.2.1. Technical data

The technical data deals with IT infrastructure, i.e. network description and associated systems, excluding physical infrastructure such as buildings. The input data is an outcome of processes implemented in T2.2, which will be enriched with metadata in the integration process, for future exploitation along the TRE$_S$PASS data management chain. In general the technical data extracted and integrated in TRE$_S$PASS covers physical and logical network infrastructure properties, namely locations, policies (IT).

3.2.2.2. Social data

In TRE$_S$PASS, social data is meant to incorporate the information regarding organisation-specific compliance behaviours, information handling processes, organisational management structures behaviours, and information security policies in practice. On the attacker side, the social data will include the actor profile and behaviours background. In practical terms, social data outcome of processes implemented in T2.3 which will be enriched with metadata in the integration process, for future exploitation along the TRE$_S$PASS data management chain.

3.2.2.3. Working with the data

In case of technical data, there are generally three ways of obtaining data. First, some types of data can be pulled out of public sources (like vulnerability databases, attack statistics). Second, some types of data can be obtained using automated extraction from the analysed infrastructure (e.g., network topology, geolocation of assets, known vulnerabilities per asset, security policies). Third, specific aspects of data need tailored

pre-processing to be used in the model. For example, information about an impact of attack might is often manually evaluated by experts.

In case of social data, data can be obtained or deduced through a number of possible ways. including: questionnaires (demographic information), police files, reports from organisations, screening instruments, narrative analysis (storytelling, scenario-based walk-through, narrative networks, etc.). Methods for collecting data and the scope of the data can be divided into data about: business data, attacks, attackers and victims.

3.2.3. Consolidating data

The process of data consolidating data from different data domains (technical, social), requires adopting specific model aggregating: definitions about attacker, attack, victim, as conceptual baseline; a data model for each of the data domains in accordance and consistent with the TRE$_S$PASS base data model. This way, the TRE$_S$PASS base model will be extended to the specific area of application, in a way that allows the integration of data from different domains into a consistent data model.

Again, the difference between the base TRE$_S$PASS data model and the extended models are the semantics or meanings which is the way data entities and properties are evaluated, since an extended model is designed to handle a specific problem or technique and its main focus is the interpretation of the concepts, while the TRE$_S$PASS model is generic to accommodate different extended models and its main focus is data structures (relationships and consistency rules) and types, and therefore, must be prepared to include general concepts that must be implemented on a real scenarios of application.

Reinforcing, the reason why TRE$_S$PASS must be able to accommodate different extended models is because different problems with different goals require different approaches.

From the WP1 model, the TRE$_S$PASS model needs to represent the following concepts (D1.1.1 (?, ?)):

- Location: related to specific domains (technical, social), and are connected through edges according to a certain access policy (which manage access to locations and data)
- Actor: may play one or more roles and interact with the infrastructure moving themselves along edges or moving data from one place to another (actor or location) by the means of actions (which describe what actors can do)
- Behaviour data: the attacker goals and attacker profiles and how those influence the security models and evaluation of the risk
- Social vulnerabilities: manipulation, social engineering, processes, etc

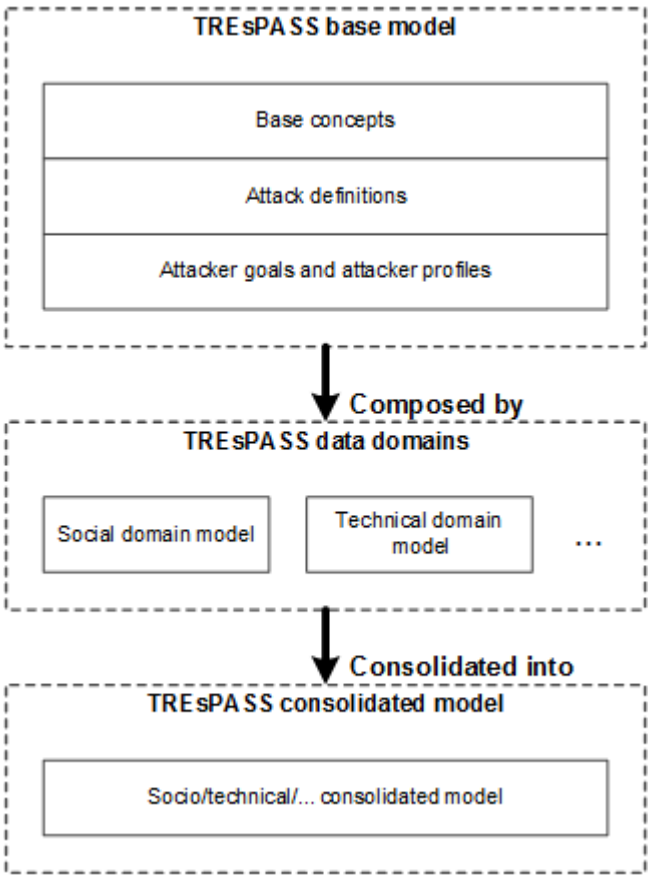

Figure 3.2.: Integration model

TRE$_S$PASS model also must represent attributes including numeric values like cost, likelihood values and probability distributions, difficulty, time, skill, etc., to be processed by other WP's or requested directly from the WP's to WP2. Then, the way how these attributes are processed and calculated (i.e. likelihood of attack depending on the context of application, can be calculated very differently), is defined by each extended model.

Finally, each data model entities must include metadata fields that allow the identification of the source domain and from where each single piece of data came. With this metadata fields it will always be possible to keep track of the data even data that has been transformed and it's not in his original format.

As said before, in order to be possible to create a consolidation process between data from different domains, data models from these domains must inherit from the same conceptual baseline model. Now, there are two ways to do that.

- From scratch – this is the cleanest and easiest way to implement the process, consisting in choosing an extended model and then developing a social and technical model based on it. This will lead the design of the data gathering processes, including social surveys, socio demographic data to extract, technical infrastructure to include, etc

- From an existing data gathering process – the fact is that most of the time, processes to gather data are already in place (for example, police records) and we want to use them, therefore, the process must be able to adapt these existing data. In this case, an intermediate model, aligned with the chosen baseline model must be developed and a data transformation process (which might be semi-automatic) between the original sources and these model must be set up. Expert judgement must be used to create this intermediate model and the corresponding data transformation process

3.3. Revision of the initial requirements

Based on the obtained knowledge during project advances, we perform the refinement of initial requirements regarding data consolidation, first introduced in D2.1.1 (?, ?). In Section 3.3.1 we outline the baseline of initial requirements and in Section 3.3.2 we discuss the validity of the requirements.

3.3.1. General baseline of consolidation requirements

In the document D2.1.1 (?, ?) we identify a set of initial requirements for the data management process. As seen, initial requirements are generic, as just the consolidation process has been identified as an important issue within the project. The following requirements relate to the data classification and validation:

- All data extraction must be conducted under the appropriate terms and conditions (from D2.1.1, Section 6.2 (?, ?))
- Methods for reliable data collection for likelihoods, costs and assets values must be identified
- Possible attack steps must be expressible in terms of comparable variables

3.3.2. Refinement of the general consolidation requirements

According to the statements described previously, the following requirements regarding data management process are extracted:

Principle WP2.6: Data discovery and extraction process must be designed in a way that enables context specific adaptations and configurations

From this requirements, a number of conclusions can be inferred, including:

- Data from different domains (technical, social, other), in order to be consolidated must be inherited from the same base TRE_SPASS model and the consistence with the model must be maintained during the inheritance process

- Data from different domains can be extracted, stored and used differently and managed independently
- However, in order to consolidate two domains, data transformation processes must exist to transform the data into the TRE$_S$PASS base model or into an intermediate data model ready to be then transformed into the TRE$_S$PASS model
- The TRE$_S$PASS data model must define base data structures (attributes, relations) and types (data must be strong types as much as possible) The semantic of the data (meaning), contain rules, data interpretation, processing and inference capabilities. In the data model, a representation of the way data is evaluated and interpreted (data meaning) must exist (data processing depends on the meaning) TRE$_S$PASS must allow using different models over the base data structures and types, done through extensions. Metadata elements used to classify data, must allow identifying the source data domain

3.4. Intermediate requirements on data consolidation

More specific requirements regarding technical and social data consolidation can be derived. The current analysis is based on the experience gained during prototype implementation and integration. The analysis and presentation of specific requirements is structured into four aspects: (a) semantic consolidation, (b) collecting and storing data, (c) data formats and access and (d) consolidation process.

3.4.1. Consolidation between domains

Documents D2.1.1 (?, ?) and D2.2.1 (?, ?) describe several scenarios where the integration of social and technical data is expected. For example, the information on the likelihood of success requires calculations that include both technical (e.g., vulnerability of an asset) and social (e.g., attacker skill, victim resistance to social engineering) data. To perform those calculations in a meaningful way, the context of different types of data needs to be adjusted (e.g., data describing the vulnerability of student population to social engineering cannot be applied in the context of technical infrastructure operated by elderly people).

Consolidation between domains means an alignment between the concepts used by the different domains (in this case technical and social). The approach proposed here to do a conceptual consolidation is to select an extended model and then use it as the baseline for the data concepts (i.e. a correspondence between concepts of the two models do consolidate). After the conceptual consolidation, a data consolidation must be also performed (i.e. data objects like attributes, types, relations and others), where data consolidation refers to the specific data collected and used.

In terms of extended models, for example, two already specified are: Crime Prevention Through Environmental Design (CPTED) (?, ?); 25 Techniques for Crime Prevention (?,

?); and others are possible. The most important aspect is to have a well-defined baseline model that can be used to derive the concepts used in the domains. This way the root concepts are common and consistency can be guaranteed between them. The TRESPASS framework must be prepared to handle different security models and not be limited to handle specific aspects of a certain one. However for each model, consistency must be guaranteed when applying and consolidating it to different domains of application. To mix different data from different domains into a consistent model, the same data model (structure) baseline must be used, otherwise consolidation might not be possible or might not produce accurate values.

So, in order to guarantee consolidation between two different domains of application, the following process must be in place:

- Build or select an existing extended model defining: attack, attacker, victim; to be used as baseline model
- Define data models for both domain of application, consistent (i.e. defining the same concepts) with the extended model chosen previously
- From there, derive the data models for both domain of applications
- Build a concepts mapping between both data models, keeping always the consistent between the extended models selected. If the previous steps were correctly executed, this one will be almost automatic if not fully automatic, and will consist mainly in cross-checking again the consistency of both data models definitions according to the extended model

There might be concepts defined in one domain of application that do not exist in the other. This in fact is the simple case, where no mapping between concepts and data need to exist. In this case what must be guaranteed is that everything is consistent with the extended model in place and then the conceptual and data consolidation is achieved through the model.

3.4.2. Collecting and storing the data

In order to be able to consolidate data from different domains (technical, social, other), data must be converted somewhere in the process. It cannot be expected that already existing data producers and consumers to be aligned with the TRESPASS models and concepts and that's the reason why data must be extracted, transformed and stored into intermediate databases to be consolidated only after that. Another relevant aspect of the data storage process is that data must be suitable to be processed by data quality (integrity check, data cleansing, correctness, completeness and data relevance) and analysis (pre-filtering, initial intelligent analysis) processes, which might require data to be organized into different data structures (prepared for aggregation techniques and other to be applied). In this case, data must be transformed again and stored into a data staging area.

3.4.3. Data formats and access

As stated before, the process of collecting and accessing social and technical does not need to be the same or even be consistent in terms of data types and data structures. The important aspect of the consolidation process is that somewhere in the process, a transformation mechanism that allows data to be represented into a format suitable to be consolidated must be executed. This means that domain specific autonomous data management processes, with independent goals and different analysis procedures, can coexist for different domains. In the limit the data consolidation process must be transparent and non-intrusive to the domain specific data management processes.

Then, the process of data consolidation must be able to consolidate the following items:

- Data type
- Context in which data was extracted
- Value and its domain
- Uncertainty on the value (interval, probability distribution, etc.)

As said previously data classification items that must be supported include: data levels of measurement, attacker data, attacker modelling, metadata elements.

Regarding data levels of measurement, TRE$_S$PASS aims to perform quantitative risk management, in the sense that countermeasures can be selected based on an estimation of the return on security investment. Therefore, the data management process needs precise definitions of what are often called levels or scales of measurement. Interpretative data, may involve elements which cannot easily be simplified to the level of a code like social and cultural factors. In terms of codified data, which express a value in a certain domain, a common distinction between nominal, ordinal, interval and ratio scales exist. The following properties apply: nominal, ordinal, interval, ratio.

As not all data can be measured precisely and representing uncertainty in the data must be possible. As a result, ordinal scales such as low, medium, high; might be used or, in alternative to use intervals to represent data. Another option will be to use probability distributions for the variables, to express the likelihood of the value being within certain limits. This is the most expressive form, but it requires some level of calculation, and representation since we might need to represent the probability function and not only the values.

Regarding the attacker data and attacker modelling, an attack navigator will have to be able to prioritize attacks and countermeasures, based on data available about the steps involved in the attack scenarios. The central requirement of the data management process is therefore the possibility to express a variety of possible attack steps in terms of comparable variables, which include: difficulty, resources spent, likelihood of success, impact of the attack.

The following attacker data is required to build the scenario, model the attacks and build the identity of the attacker: definition, motivation, capabilities, knowledge about the organization, strategy, resources.

Finally, regarding metadata elements, the following must be included when classifying TRESPASS data: security classification, reliability, provider, date collected, source domain, durability, owner.

3.4.4. Consolidation process

While the technical data covers the topology of the security network, social data will cover the methods and behaviours that can be used to move across the security network.

These two types of data are conceptually different and these differences will be reflected in different extraction, classification and validation techniques to be used for the different data types. The result of the extraction, classification and validation process will be stored in a common model and both types of data be consolidated in a common database.

Being in a common database after this consolidation process both types of data can be seamlessly handled by the unified data management process.

The following steps are possible to implement a consolidation process:

1. Identify already existing domain specific data processes (technical, social, other)
2. Identify the extended model used by the social data process, align with TRESPASS concepts. If none is present adapt one that matches the social data model (export judgement might be required)
3. Check if the technical data processes fits the extended model used by the social data processes. If not the model must be adapted or a new one must be chosen
4. Develop one new data model for each of the data domains consistent with the extended model and the TRESPASS concepts
5. Develop data transformation processes to convert original data from the source data domains model into the new models and use them to load data into the models
6. Apply consistency rules to the resulting data model loaded with the data imported

This can be summarize into a single principle, which will be used to develop WP2 requirements, together with the other WP's requirements:

Principle WP2.7: Data from different domains (technical, social, physical, etc.) data must be available in a consolidated data model to handle all data seamlessly and in a standardized way

From this principle, a number of conclusions can be inferred, including:

- After the consolidation process both technical and social data must be available in the consolidated data model, an extension to the TRESPASS base model, allowing the unified data management process to handle all data seamlessly

4. Handling external data sources

In this chapter we analyse two aspects of data integration: (a) integration with data from other WPs and (b) integration with external sources of information.

The process of data integration includes combining data from different data sources and providing data consumers with a unified view of these data. In order to achieve this goal, a number of data processes must be set up, including a global data schema definition and the ability to perform data transformations between the data sources schemas and a global data schema. Then data connection and communication processes must also exist.

4.1. Progress on handling external data sources

The data integration model adopted in TRE$_S$PASS will be based on the data integration theoretical perspective (?, ?), where three main concepts must be defined:

- A global schema
- The sources schemas
- The mappings between the sources and the global schema

Data mapping can be defined as the process of creating data element mappings between two distinct data schemas to be used in the data integration process. There are two major ways to model the correspondence between the global and source schemas (mapping): Global as View (GV) and Local as View (LV).

- In the GV approach, the mappings associates to each element of the global schema as a query over each source schemas, making this method more suitable when the local data sources are unlikely to change
- On the other hand, in LV approach, the mappings associates to each element of each source schemas a query over the global schema, making this method more suitable when not all the data sources are known and new ones will be added over time, and the global schema is more stable and unlikely to change

In the context of the data integration process of TRE$_S$PASS, a LV approach will be favoured.

In order to define a global schema, currently there is an important project activity under execution from **Task T6.2: Refinement of functional requirements**, *"[...] define the*

architecture, including the database framework and data model (standardization in XML schema), to allow interoperability of different components implemented in the previous WPs [...]".

Finally, and since not all tool or applications, and specially data sources, support or will support the TRE$_S$PASS XML schema format, associated to the XML schema local data mappings (transformation processes) must be defined for each case, that allows transforming data from different formats into the TRE$_S$PASS XML schema. In order to achieve that, a data process will be defined and the implementation of each data mapping performed locally for every tool, application or data source.

This can be summarize into a single principle, which will be used to develop WP2 requirements, together with the other WP's requirements:

Principle WP2.8: It must be possible to transform different schemas types and define local data mapping per data source

From this principle, a number of conclusions can be inferred, including:

- It must be possible to transform between data source schemas and global data schema
- It should be defined a local data mapping per data source (usually an ETL: extract, transform, load; process, specially for technical data, since for social data, this industry standard methodology might not be enough, because of the nature of the social data)

4.2. Revision of the initial requirements

4.2.1. General baseline of integration requirements

In the document D2.1.1 (?, ?) we identify a set of initial requirements for the data management process. The following requirements relate to the integration of external actors:

- **IR2.7:** The audiences and purpose of data extraction must be identified prior to extraction
- **IR2.8:** Data discovery and extraction process must be designed in a way that enables local adaptations and configurations
- **IR2.9:** All data extraction must be conducted under the appropriate terms and conditions (from D2.1.1, Section 6.2 (?, ?))
- **IR2.10:** The different external sources should be categorised and investigated (IDS, penetration testing, social engineering etc)

4.2.2. Refinement of general integration requirements

All the data integration requirements from the previous deliverable remain valid. In this deliverable the objective is not only to develop those existing requirements but also to infer new ones.

4.3. Requirements on actor integration

In order to perform a data integration process, a transform process must exist to convert each local data schema into the global one. The transformation process applies a series of rules and functions to the source data to derive target data schema. Some data sources require very little or even no manipulation of data. In other cases, one or more of the following transformation types may be required to meet the final data schema:

- Multiple data sources joining (lookup, merge)
- Data selection
- Coded values translation and values encoding
- Calculated value computation
- Data sorting
- Data aggregation
- Data splitting or joining (one data element split into several elements or vice versa)
- Repeating data disaggregation (cleaning, duplication removals and consolidation)
- Transposing or pivoting (turning multiple columns into multiple rows or vice versa)
- Surrogate key generation to relate data
- Business/semantic rules appliance
- Data integrity validation (relations, valid values, existence, unique, etc.)
- Data deduction (when not enough data exists, default rules or values must apply)

A complete data transformation process must be able to handle all of these transformation methods, being the data deduction the one where the level of uncertainty might be higher. However not all of these methods will be required in every specific data transformation that must be applied. Different technologies might be used to accomplish the data transformation process. In the end, the final goal is to have the data according to the global data schema, which can be achieved by using different technologies depending on the scenario, source data types, knowledge of the technical personal, etc. For example the XML based XSLT technology might be good to handle certain transformation methods (i.e. transposing or pivoting), but inappropriate or unable to handler other ones (i.e. multiple data sources joining, so a special attention must be paid when choosing the technologies or tools to support the process.

The scalability level of the extraction and transformation process must be established across the lifetime of its usage. This includes understanding the volumes of data that must be processed within service level agreements, which may change over time, as the processing times may also change.

This can be summarize into a single principle, which will be used to develop WP2 requirements, together with the other WP's requirements:

Principle WP2.9: There must be a global data schema defining the TRE$_S$PASS concepts like location, actor, behaviour data, social vulnerabilities, numeric variables, and others, defined under WP1 model

From this principle, a number of conclusions can be inferred, including:

- There must be a global data schema defining TRE$_S$PASS concepts. These schema is used to validate data contents according to its structure, and there must be a way to validate a data content against the schema. Data defined according to the global data schema definition must be suitable to be transformed into other data schemas through a mapping process, as long as the data base concepts remain the same. Data contents structured according to the global data schema, must be independent to any data transport protocol, which means that different transport protocols can be used to transmit TRE$_S$PASS data, and no dependency must exist between the message format and the transport protocols
- The TRE$_S$PASS global schema needs to represent the following concepts:
  - Location: related to specific domains (technical, social), and are connected through edges according to a certain access policy (which manage access to locations and data)
  - Actor: may play one or more roles and interact with the infrastructure moving themselves along edges or moving data from one place to another (actor or location) by the means of actions (which describe what actors can do)
  - Behaviour data: the attacker goals and attacker profiles and how those influence the security models and evaluation of the risk
  - Social vulnerabilities: manipulation, social engineering, processes, etc
  - The model must represent attributes including numeric values like cost, likelihood values and probability distributions, difficulty, time, skill, etc

4.4. Interactions with other WPs

What are the interactions with the other WPs, namely WP1, WP3, WP6. Describes how WP2 data management process can increase the value of the data that is delivered.

4.4.1. Data schema definition

The objective of this activity is to develop a data XML schema to be used as a contract for data exchange between TRESPASS tools and applications. This XML schema defines and describes certain types of XML data by using an XML Schema definition language that guarantees consistency among certain types of XML data that is shared between the TRESPASS tools. In short, XML schema elements (elements, attributes, types, and groups), which compose the XML schema document are used to define the valid structure, valid data content, default values and relationships of certain types of XML data. Key aspects of the XML schema design are:

- To support reusable types and allows the creation new ones (preferable by using inheritance)
- To group elements to control the recurrence of elements and attributes

In order to achieve this activity a process was defined and is being under development, which includes the following sub-activities enumerated:

1. Make a survey about the needs from other stakeholders regarding the model.
2. Aggregate and consolidate the work that is on progress by several stakeholders related to data modeling, including identify stakeholders and contributors
3. Validate TRESPASS concepts (actors, locations, policies, etc.), and relations between them.
4. Validate metadata associated with the information
5. Create a data model aligned with the concepts and metadata and build an XML schema according to that

As inputs to create the XML schema, the following items were taken into consideration:

- TRESPASS glossary of common concepts
- WP2 conceptual data model and concepts (from D2.2.1 (?, ?))
- Initial TRESPASS datamodel (derived from D2.2.1 (?, ?))
- Other WP deliverables with data representation information (mostly WP1, WP2, WP3, WP6).
- ADTool specifications (from D1.3.1 (?, ?))

The subsequent steps of the process of building a TRESPASS XML Schema include:

- Develop conceptual data model (iterate through D2.2.1 (?, ?))
- Create an SQL data model to support TRESPASS concepts (linked with other WP2 and WP6 tasks)
- Convert the SQL data model into an XSD Schema (or generate a "complete" XML sample and convert it into the XSD Schema)

- Convert the XSD Schema into a DTD Schema in order to support both formats

This can be summarize into a single principle, which will be used to develop WP2 requirements, together with the other WP's requirements:

Principle WP2.10: A data XML schema should be developed to be used as a contract for data exchange between TRESPASS tools and applications.

4.5. Interaction with external sources of information - project progress

This section describes the project progress in terms of data integration with external actors and sources of information, including data model design and formats, decisions and tool planning.

Data transformation is used to mediate the relationship between an initial data source and the destination in which the data is used. It is useful in identifying data elements and the way in which data flows. This can be divided into two parts:

- Data mapping, the process by which two distinct data schemas are created and linked. Those data schemas can include metadata, and the mapping also includes discovering hidden information and sensitive data such as social security information that must be hidden or obfuscated (data masking). Mapping is frequently complicated by complex transformations that require one-to-many and many-to-one transformation rules
- Data transformation, the program that runs the transformation, using available technologies including Java and XSLT (popular for XML transformations)

So, the most relevant aspects of data interaction are data mapping and transformation. When building a software tool or module, these features are critical to allow using TRESPASS data, and the concept of data adapter (software modules that handle data mapping and transformation) must be introduced. This way, before using it, every tool data must be passed first to a number of processing steps that manage the process. A TRESPASS data manipulation tool must then include three main components:

- Data input module: the entry point that connects of received data from external source which initiates the entire data manipulation process. The entry point can be implemented by an endpoint waiting to be activated through external calls (i.e. webservice), which passes the execution context to the initial handler where the external input data received from the data source is read and an initial data analysis executed to read input call parameters (not data content), logged and content data put into the internal data objects. This module should be composed by the following sub-modules:
  - Source reader transport (mandatory) – reads physical input data stream, reading data from the input data source

  - Source reader format (mandatory) – reads and interprets data, converting the external data format into the internal data structures, in order to be manipulated by the additional modules or sub-modules
  - Data mapper decoder (optional) – reads data received in order to connect it with previous data collected ((if more than one invocation call is involved, which is important when the process includes multiple calls used to update or complete data already collected, meaning that, in this case a data storage must be set up), or in order to create a new data set of data. This is where data mappings are performed and stored into internal data objects, so this sub-module might include a lot of application logic
  - Logger (optional) – stores original data received and associated metadata, like timestamps, origins, etc
- Data manipulation module: interprets, manipulates and logs internal data, through the execution of one or more modules which implement specific well-known data concepts (location, actor, policy), and general logic. Since all kinds of data manipulation is possible no mandatory processing sub-modules must exist. Nevertheless some possible examples are:
  - Data expansion (optional) – reads all data set data for missing records and expand those missing records whenever possible (i.e. a policy that references an actor that doesn't exist, which leads to the creation of a new actor). When no information exists to expand missing records, warnings must be generated or the data set must be marked as invalid
  - Data validator (optional) – validates all data set for consistency between concepts and marks the data set as valid or invalid according to the validation criteria implemented
  - Data packaging (optional) – consolidates all data set into a single message adding additional information as needed (i.e. headers and specific configuration data)
- Data output module: reads internal data objects, logs and creates an external data output to be sent to the data target
  - Target writer transport handler (mandatory) – writes physical output data stream, sending data to the output data target
  - Target writer format (mandatory) – reads and interprets internal structures format data, converting it to an external data format, in order to be sent to the output data target
  - Data mapper encoder (optional) – encodes data from the internal objects into the target data structures. Does the opposite logic of the data mapper decoder defined previously Logger (optional) – does the same as previous, but now with the output data

Additional optimal modules of the data mapping and transformation process implemented by the tools might also include:

- Data staging area – stores data into a local persistence data model, including well-known data concepts (location, actor, policy). A key must be used to connect well known data concepts from a single data set. With this feature, data consumers can perform several analyses and adapt results as the environment changes over time
- Data query – provides a number of services to query data from the data model, from general query service to specific well-known data concepts (location, actor, policy)

This can be summarize into two principles, which will be used to develop WP2 requirements, together with the other WP's requirements:

Principle WP2.11: TRE$_S$PASS data tools must implement a TRE$_S$PASS data management process

Principle WP2.12: Data stored in the TRE$_S$PASS platform must have parameterised life cycles and secure destruction mechanisms

5. Future developments and planning

5.1. Next steps

Still, the work currently in progress needs further develop which will be the motivation for the succeeding deliverable. Major topics to be developed are:

- Data Design by developing data formats and semantics to allow technical/social data consolidation, including data structures and process
- Data quality process by assessing the value of the factors that comprise the data, like integrity check, data cleansing, correctness, completeness and data relevance
- Data pre-analysis process by offers pre-processing capabilities over the data introducing an analysis layer that adds value to the data
- Data Integration by combining data residing in different sources and providing a unified view of these data

Other relevant topics, with a higher level of detail, include:

- Working and understanding technical issues related to scalability
- Prepare tools architecture and design, including the idea of data connectors (for data transport and format)
- Understand how data can be managed in a way that environmental changes must also change the way data is processed
- How can the process support the development of business cases for those institutions suitable to become TRESPASS final users

5.2. Schedule

During year 3 of the project it is predicted that the above developments are finalised and the final requirements closed. A first version of the specification of the information system is provided.

During last year of the project Data Uncertainty Management and testing and degradation options will be further developed.

5.3. Risks and uncertainty

The data management framework should be improved and adapted over time. For example the subtopic: Unknown, undefined and estimation; might be promoted in the future to a major topic like: Data Uncertainty Management; which will have a delivered by its own in the future (?, ?), as the discovery process advances.

6. WP2 Requirements List

This document introduces a number of principles used to guide WP2 (data management process) requirements elaboration. In order to accomplish this objective, a synchronization with all other WP's was necessary in order to deliver a TRE$_S$PASS consolidated requirements list, avoiding requirement overlapping or inconsistencies. Process can be summarized as:

1. Develop WP2 principles;
2. Publish principles as WP2 requirements in TRE$_S$PASS requirements central repository;
3. Consolidate WP2 principles with other WP's requirements in order to deliver a final and consolidated WP2 requirements list avoiding inconsistencies and duplications between WP2 requirements and requirements from the other WP's;
4. Publish the final requirements list (requirements from WP2 and to WP2) in this deliverable.

As a result, the final consolidated requirements list for WP2, is enumerated bellow.

6.1. Requirements from WP2 to other WP's

Requirement R09

Requirement : Classification of data types

Source WP: WP2

Target WP: WP4

Goals: Needed to produce visualisation toolkit

Acceptance criteria: Visualisation toolkit supports XML model data types

Status: Agreed

Dependencies: Trespass XML model to be defined. The ability to accurately define the data types will depend on the quality of data that comes through from WP2.

Requirement R19

Requirement : Socio-technical security models must provide the necessary abstraction to represent infrastructure and other components.

Source WP: WP2,6

Target WP: WP1

Goals: Needed to use the model

Acceptance criteria: XML Format defined

Status: Agreed

Dependencies: None

Requirement R41

Requirement : Data categorisation needed

Source WP: WP2

Target WP: WP2,7

Goals: Require categorisation of data types and capabiltiies

Acceptance criteria: Trespass Model and Attack Pattern Library data requirements specified

Status: Agreed

Dependencies: All data types from the different tools chains are clarified

6.2. Requirements from other WP's to WP2

Requirement R01

Requirement : Common XML shared model

Source WP: WP6

Target WP: WP2

Goals: simplify data exchanges between tools

Acceptance criteria: Model available for current toolset

Status: Agreed

Dependencies: None

Requirement R02

Requirement : Central database structure

Source WP: WP6

Target WP: WP2

Goals: store data needed by several tools

Acceptance criteria: Structure available for current toolset

Status: Agreed

Dependencies: None

Requirement R08

Requirement : Parameter values for the trees in APL

Source WP: WP5

Target WP: WP2

Goals: Usability of APL depends on it. Needed to produce meaningful analysis results.

Acceptance criteria: Parameter values are provided and look reasonable

Status: Agreed

Dependencies: None

Requirement R14

Requirement : Develop visual thinking tools

Source WP: WP4

Target WP: WP2,4,5

Goals: Needed to produce meaningful analysis results

Acceptance criteria: Documented thinking tools accepted by academic publication peer reviewers (achieved), documented thinking tools accepted by project reviewers in deliverable 2.3.2 (on-going) tools accepted by practitioner panels (iterative and on-going)

Status: Agreed

Dependencies: None

Requirement R15

Requirement : Define methods of visualisaiton evaluation

Source WP: WP4

Target WP: WP2,4

Goals: Needed to ensure robust tool

Acceptance criteria: Documented in deliverable 4.2.1 and reviewed by project reviewers (achieved) visualisations accepted by practitioner panel (iterative and on-going)

Status: Agreed

Dependencies: None

Requirement R16

Requirement : Provide model content such as infrastructure, policies, etc

Source WP: WP1

Target WP: WP2

Goals: Needed for model

Acceptance criteria: For the case study scenarios, the relevant elements and their properties are provided as text documents, and enable a modeller to create a matching WP1 model.

Status: Agreed

Dependencies: None

Requirement R28

Requirement : TRESPASS tool will need to be handed over to one or several industry partners for testing and application.

Source WP: WP7

Target WP: WP2

Goals: Needed for the models, in relation to compatibility of data formats

Acceptance criteria: The tool should consider compatiblity with different platforms.

Status: Agreed

Dependencies: None

Requirement R29

Requirement : TRESPASS tool needs to provide performance scalability in order to (potentially) be able to cope with massive amounts of data, e.g. billions of call data records (CDRs).

Source WP: WP7

Target WP: WP2

Goals: Needed for the models

Acceptance criteria: The tool should be scalable.

Status: Agreed

Dependencies: None

Requirement R30

Requirement : The data formats used in the tool should be flexible as data sources in the domain of the respective industry partners could be different.

Source WP: WP7

Target WP: WP2

Goals: Needed for case studies

Acceptance criteria: The TREsPASS tool should accept known data fomats (XML, JSON, other)

Status: Agreed

Dependencies: None

Requirement R41

Requirement : Data categorisation needed

Source WP: WP2

Target WP: WP2,7

Goals: Require categorisation of data types and capabiltiies

Acceptance criteria: Trespass Model and Attack Pattern Library data requirements specified

Status: Agreed

Dependencies: All data types from the different tools chains are clarified

Requirement R43

Requirement : The model and data extraction must be able to cope with a dynamic system, i.e., changes in the system need to be detected and represented in the model. And the model should have rules to change the model itself.

Source WP: WP7

Target WP: WP1,2

Goals: Needed for cloud case study, since the systems are highly dynamic.

Acceptance criteria: The data extraction tools need to detect and obtain change events in the cloud infrastructure. Those change events need to be translated into a change for the model and applied to the model.

Status: Agreed

Dependencies: None

Requirement R63

Requirement : The system should be able to supply data on hardware, software, version, configuration, as well as social aspects, upon request of the user when building or editing a model.

Source WP: MT

Target WP: WP2

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 1 (Security Investment), specifically U1.14.

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

Status: Agreed

Dependencies: R5

Requirement R65

Requirement : The system should be able to autonomously collect public data for future use in models, by updating component / attacker libraries.

Source WP: MT

Target WP: WP2

Goals: Derived from P1 (TRESPASS tools should assist the user in building a navigator map). Supports Use-case 1 (Security Investment) and Use-case 4 (Product-service system), specifically U1.13 and U4.2.

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

Status: Shelved

Dependencies: None

Requirement R84

Requirement : The TRESPASS analysis should suggest a ranked list of countermeasures, with associated map components which can be dragged-and-dropped into the model and affect the analysis accordingly. The countermeasures should be available in a library.

Source WP: MT

Target WP: WP1,WP2,WP3,WP4

Goals: Supports Use-case 2 (Audit), Use-case 4 (Product-service system) and Use-case 5 (Quick scan), specifically U2.10, U2.12, U.4.2, U5.3, U5.13, U5.14.

Acceptance criteria: 80perc. of the users specified in the Use Cases are satisfied with the suggested countermeasures. 80perc. of users are able to successfully include counter- measures them into the model.

Status: Shelved

Dependencies: None

7. Conclusions

Requirements mean needs that a particular design, product or process must be able to perform, which, in this case refer to the TRESPASS data management process.

In D2.1.1 - Initial requirements for the data management process, high level requirements have been introduced (?, ?).

With the prototype development and implementation in D2.2.1 - Technical data extraction prototype (?, ?), the initial requirements have been grouped and based on its existing requirements have been refined and new requirements identified. This document presents the outcome of the work performed so far. Other relevant requirements arise while trying to adopt the knowledge developed in TRESPASS to real world problems (not limited to the project case studies).

From the initial revision several requirements have been refined, for example requirements related to data consolidation and output processes.

In this deliverable D2.1.2 - Final requirements for the data management process, we focused on developing the requirements and introducing other aspects of the data management process required by the TRESPASS project. This includes requirements for the technical and social consolidation, integration with external actors.

Data management process requirements were internally divided into four areas of knowledge: data design, data integration, data quality, data analysis; which agregate the requirement into more granular pieces of information.

A. Project Summary

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

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

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

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

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

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

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

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

The innovative visualisations (WP4) developed in TRE$_S$PASS focus focus particularly on visualising elements of the analysis, as this is key to the overall project goal of providing "decision support" to practitioners. However, visualisations contribute also to model development and data gathering.

Practitioners can access the TREsPASS toolkit via the attack navigator map interface, which provides an intuitive means of selecting appropriate tools (WP6) for data gathering, modelling, analysis and visualisation. These can be used, individually or in combination, to strengthen operational and strategic decision-making.

A.1. Case Studies

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

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

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

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

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

A.2. Overview of TRESPASS Integration

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

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

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

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

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

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

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

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

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

- **Components customization (optional)** takes place before or during the TRESPASS model creation to create specialized custom model components.
- **Attacker profile creation** creates the attacker profile that the TRESPASS model analysis should consider, based on ready-made attacker profiles.
- **Defender/target profile creation** creates similar profiles for the other actors in the model based on the social data gathered in the social data collection activity.

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

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

1. In the **attacker profile selection**, the user selects the attacker profile to use in the analysis.

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

2. The **attacker goals creation** provides the attack generation with the attacker goals. These can be derived by hand from the stakeholder goals or deduced automatically from the selected attacker profiles.
3. The **scenario selection** selects a scenario, consisting of a single pair of attacker and attacker goal, to run the TRESPASS analysis on.
4. To extend attack trees, **attack pattern creation and sharing** provides libraries with known attack steps. The attack tree generation can only reach a certain level of abstraction, which may not be sufficient for quantitative analyses.
5. **Attack generation** transforms the TRESPASS model to an attack tree.
6. **Attack tree annotation & augmentation** then extends the attack tree with attack patterns and decorates leaf nodes with parameter values from the data collection stage for quantitative analysis.
7. The **attack tree analyses** compute quantitative properties of attacks, *e.g.*, utility for the attacker or success probability of the attack.

The analysis of the **e3value model** is complementary to the core TRESPASS analysis and has only one step:

1. For the **fraud model generation**, the user needs select an attacker and an interval of expected occurrence rates of the commercial transactions specified by the e3value model. The e3fraud tool then identifies all possible violations of contracts, the loss for actors, and the delta in profit for the other actors.

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

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

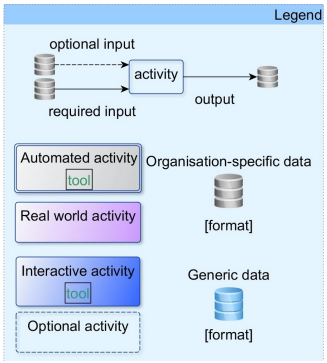

Figure A.1.: Legend for the Integration diagram in Figure A.2.

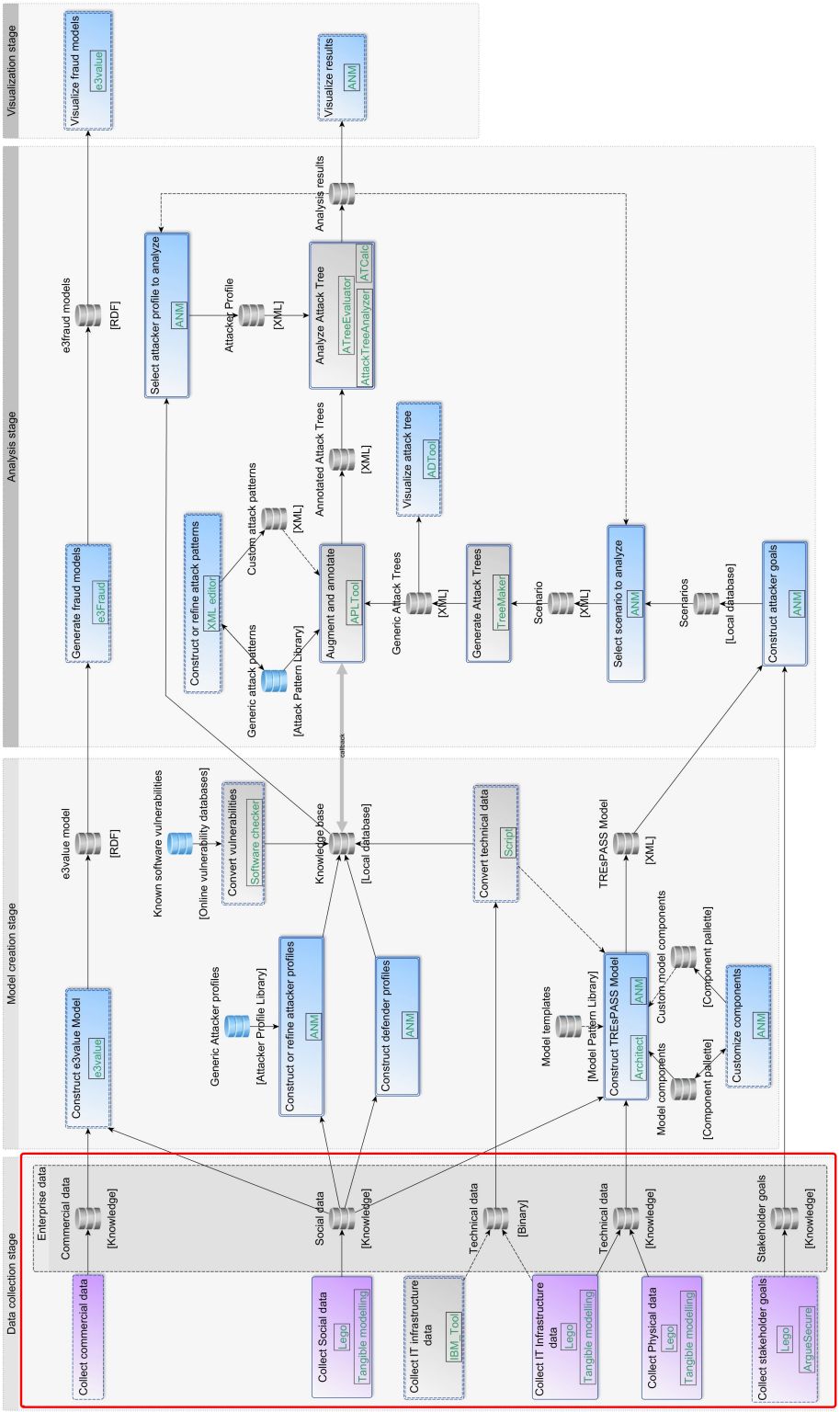

Figure A.2.: Integration diagram for the TRESPASS project.