TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Informatics
Chair of Information Systems

Integration of the TREsPASS Toolset and the ISKE Tool

Bachelor's Thesis

Student:	Vlada Plaskovitskaja
Student code:	123902IABB
Supervisor:	Aleksandr Lenin, M.Sc.

Tallinn
2015

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

(*kuupäev*)

(*allkiri*)

ISKE ja TREsPASS rakendustööriistade integreerimine

Annotatsioon

Bakalaureusetöö teema on „ISKE ja TREsPASS rakendustööriistade integreerimine“. Töö eesmärk on luua rakendust, mis täiustab ISKE rakendustööriista analüüsivõimet TREsPASS rakendustööriistade abil, pakkuda modelleerimise ja edasiarendamise nõudeid ISKE rakendustööriistale.

Et selle probleemi lahendada mina õppisin ISKE ja TREsPASS rakendustööriistu, TREsPASS mudeli formaadi ja andmetüüpe, lõin klassid andmete hoidmiseks ja funktsioonid nende andmete salvestamiseks ISKE rakendustööriistasse, üleslaadimiseks ISKE rakendustööriistast ja XML formaadis salvestamiseks, realiseerides seda funktsionaalsust iseseisvas tarkvaras. Programmi loomisel leidsin palju nüansse, mis hiljem esitasin ISKE rakendustööriistale modelleerimise nõuete nimekirjas.

Minu töö tulemuses olen loonud programm, mis integreerib TREsPASS ja ISKE rakendustööriistu ja vastupidi, pakkusin modelleerimise nõuete nimekirja ISKE rakendustööriistale ja ISKE edasiarendamisele nõudeid.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 43 leheküljel, 7 peatükki, 15 joonist, 1 tabel.

Töös on kujutatud: ISKE rakendustööriista ekraanipiltide visandid, Microsoft Visio 2013 ja IBM Rational Rose programmides tehtud kujundid ning loodud programmi ekraanipildid.

Abstract

The topic of the thesis is „ Integration of the TREsPASS Toolset and the ISKE Tool “. The goal of the thesis is to create a program that enhances the ISKE Tool with analytical capabilities of the TREsPASS Toolset, to suggest a list of modeling requirements and further development requirements for the ISKE Tool.

For solving this task I have studied the ISKE Tool and the TREsPASS Toolset, I learned the TREsPASS model format and data types, created classes for storing this data, created functions for uploading this data into the ISKE Tool, exporting from ISKE Tool and storing in XML format, implemented this functionality in the form of the stand-alone piece of software. As a result of decisions taken on the design of the program it was found lot of subtleties, which were later presented in the form of requirements for modeling with the ISKE Tool.

The result of the thesis I created a program that integrates TREsPASS Toolset into the ISKE Tool and the opposite way, I propose a list of modeling requirements for the ISKE Tool and further development requirements for ISKE.

The thesis is in English and contains 43 pages of text, 7 chapters, 15 figures, 1 table.

In the work depicted: ISKE Tool screenshots, figures made with the Microsoft Visio 2013 and IBM Rational Rose and created program screenshots.

Glossary of Terms and Abbreviations

ISKE	Three-level IT baseline security system [1]
TREsPASS	***Technology-supported Risk Estimation by Predictive Assessment of Socio-technical Security*** [2] research project that develops methods and tools to analyze and visualize information security risks in organizations, including possible countermeasures.
XML	***eXtensible Markup Language*** [3] XML was designed to describe data.
EU FP7	***European Union 7 th Framework Programme for Research and Technological Development*** [4] Indeed, FP7 is a key tool to respond to Europe's needs in terms of jobs and competitiveness, and to maintain leadership in the global knowledge economy.
SQL	***Structured Query Language*** [5] SQL is a standard language for accessing and manipulating databases.
UML	***Unified Modeling Language*** [6] UML is an industry standard modeling language with a rich graphical notation, and comprehensive set of diagrams and elements.
BSI	***In german, Bundesamt für Sicherheit in der Informationtechnik; in english.*** [1] Federal Office for Information Security
SME	***Small and medium-sized enterprises***
DOM	***Document Object Model*** [7] The Document Object Model provides APIs that let you create, modify, delete, and rearrange nodes.

SQLite JDBC ***Java Database Connectivity*** [8]

is a library for accessing and creating SQLite database files in Java.

CIA triad ***confidentiality, integrity, availability*** [9]

three main security goals

Acknowledgement

The material used in this thesis is part of the EU FP7 project TREsPASS (ICT-318003). This thesis contains only the author views and the Union is not liable for any use that may be made of the information contained herein.

List of figures

List of tables

Table of contents

1. Introduction

The modern computer world represents various and very difficult set of computing devices, data processing systems, telecommunication technologies, the software and means of its design. The entire interconnected system solves a huge range of problems in different areas of human activity, from the simple solution of school tasks on the home personal computer to management of difficult technological processes.

The more important is the area, in which computer information technologies are used, the more and more critical are such properties as reliability and security of information resources. Malware impacts information carried out to violate the confidentiality, integrity and availability. The solution of the tasks related to the prevention of damage on information is carried out within a complex problem of information security and has well-developed scientific and methodical base.

Malicious attacks result in damage to the affected parties. This damage can be either indirect or direct. Direct damage can be, for example, theft of any business secret of the company because the company will lose part of the market. Indirect damage may be an affected reputation, loss of clients, bankruptcy. Absolute information protection is impossible, that is it is impossible to protect the system so that it could not be attacked. The goal of security is to minimize possible damage. All security measures cost money. Therefore it is necessary to optimize investments into security, that is to reach the greatest possible efficiency of protection spending as little as possible money for security. Often in real life the budget is limited and it is necessary to find the maximum and the most optimum set of security measures which can be deployed considering this budget. Investments into information security have a practical upper bound – it is not rational to invest into security more than the value of the protected information or assets.

Attacks can occur at the different dimentions:

- The physical (assault, blackmail, bribery)
- Cyber / the technical (for example, connecting to a surveillance camera for reconnaissance purposes)
- The social (impersonation, pretext)

Real-life attacks are as a rule complex multi-step attacks where different attack steps belong to different dimensions.

Lately software security started playing a very important role, because many attacks are carried out against programs. If the system does not have the means to detect and block access to outsiders, it can lead to a successful action of hackers and serious consequences. For example, in October 2014, hackers were able to steal data on 83 million accounts belonging to customers of the bank JPMorgan. As a result of hacking stolen data related to 76 million accounts of individuals and companies 7 million accounts, representing small and medium businesses. This hacker operation became the largest attack on banks for all history. According to The New York Times, the attackers got access to 90 bank servers and all the software. Attack of such level suggests that financial institutions are required to strengthen the level of protection against cyberattacks even more. Hacker threat compels banks to look for new ways on its prevention. [10]

There are many more such examples of successful actions of attackers. The effective solution to this problem is to analyze the threats which the organization can be subject to, and deploy the security measures in an rational (optional) way.

1.1 Background and the problem

The TREsPASS project aims at creating a software toolkit for analyzing and prioritizing information security risks. To evaluate the project and receive a benefit from it, it must be applied in practice. In Estonia the most evident place where it can be applied is ISKE. It is a three-level catalog of threats and countermeasures, the implementation of which is necessary to achieve and maintain security of information systems. ISKE Tool makes it easier to work with the ISKE standard which was developed for the Estonian public sector and is mandatory for all state and local government information system databases. ISKE Tool generates only threats and protection measures and does not prioritize information security risks, to a cost-effectivnes comparison of available security measures. Combined with the TREsPASS Toolset, it will be a more efficient solution for preventing information security threats. TREsPASS analysis system will be a useful supplement to the ISKE Tool, because it points out the weaknesses in the system and most important security measures that should be implemented right now. Differently from the ISKE Tool, the TREsPASS Toolset performs near real-time analysis. It is very important because in an average and large infrastructure there is a highly dynamic threat landscape, which constantly changes. Every day something develops, new technologies and new vulnerabilities emerge therefore we cannot analyze the system once and claim that it is safe. Security is a process, not the state and thus the system should be analyzed constantly. The TREsPASS Toolset allows it, while the ISKE has a long update cycle and does not change in the short run. Therefore the integration of the TREsPASS Toolset into the ISKE Tool is one of the application objectives of this project and I hope that by means of the TREsPASS Toolset the ISKE Tool will be improved. Also with the help of prioritization of security risks in the TREsPASS Toolset it will allow to receive the greatest efficiency of protection spending reasonable amounts of investments on protection. Integration in the opposite direction is important as well, because ISKE Tool is mandatory in Estonia. In this case the analysis starts with the TREsPASS Toolset from which an ISKE model with the list of corresponding diffensive measures can be produced later on.

The Bachelor's Thesis "Assessment of integration possibilities of the TREsPASS toolset into the ISKE Tool" [11] has already given a short overview of the integration possibilities of the TREsPASS Toolset into the ISKE Tool - as a result of this work it became clear that the integration of products in the form as they are now is impossible. The author concludes that *"such data like processes and policies cannot be provided on such level of detalization, which is required by the TREsPASS toolset. The integration requires more effort and creation of additional external components."* [11]

This thesis deals with the next step - making a program that integrates the TREsPASS Toolset into the ISKE Tool and the other way around. One of the outcomes of this work is a list of modeling and further development requirements for the ISKE Tool.

1.2 Objectives

The objectives of this thesis are the following:

1. To create a program that integrates the TREsPASS Toolset into the ISKE Tool and the opposite way.
2. To suggest a list of modeling requirements for the ISKE Tool.
3. To give a further development requirements for the ISKE Tool.

1.3 Methodology

A program that would integrate the TREsPASS Toolset into the ISKE Tool and the opposite way is created using the Java language. As a starting point I take an example of the TREsPASS project-internal case study provided by IBM Research (Zurich) and the corresponding TREsPASS model [12]. Based on this I designed the central component of an inegration solution called the integration governance component which stores all the data. Proceeding from the case study data I have detected the subset of data, which makes any sense in the ISKE model and the ways how it can be expressed there. The remaining data, such as processes and policies did not fit into the ISKE model and I placed it in an external library. Therefore I created classes for storing information related to the TREsPASS model and required functions for transformations between the ISKE and the TREsPASS models. These functions allow to convert this data into an ISKE model, as well as into TREsPASS model. I implemented this functionality in the form of stand-alone piece of software.

As a result of decisions taken during the design of the program a lot of subtleties were found, which were later presented in the form of modeling requirements using the ISKE Tool. Based on the results I suggest further development of the requirements for the ISKE Tool to facilitate even more tight integration of TREsPASS and ISKE.

1.4 Thesis overview

The thesis consists of 6 sections.

Section 1 sets the scope of the thesis and outlines its objectives and methodology.

Section 2 describes the TREsPASS navigator map and the TREsPASS workflow.

Section 3 outlines The ISKE goals, structure, shortcomings, the ISKE Tool and compares ISKE to TREsPASS.

Section 4 describes TREsPASS model integration into the ISKE Tool and vice versa, required data types, data transformation procedures and information on how model components are connected with each other, as well as what programming languages and technologies used for development.

Section 5 contains requirements for the modeling process aiming at creation systems which can serve as input for the TREsPASS analysis platform.

Section 6 contains description and screenshots of my solution.

2. TREsPASS

TREsPASS is a research project on Technology-supported Risk Estimation by Predictive Assessment of Socio-technical Security, which is funded by the European Union.

The TREsPASS project develops methods and tools to analyze and visualize information security risks in organizations, including possible countermeasures.

"We build "attack navigators" to identify which attack opportunities are possible and most pressing, and which countermeasures are most effective." [2]

In order to understand the integration solution described it the thesis it is necessary to outline the TREsPASS workflow and the structure of the navigator map.

2.1 Navigator Map (formal description)

Navigator Map is the central component of the TREsPASS Model.

It contains the following entities:

Locations

The main element in the model. It is where the actors, data or items can be. Location may be in a physical or in a cyber-zone. Zones are used to limit the boundaries of possible movement of actors and data. Location contains the following fields:

- id – a model-wise unique string
- domain – [physical, cyber]
- atLocations – optional string

For example, location ID may be an office and domain – physical or location ID – user's data and domain – cyber.

A single location may be connected with other locations, using edges.

Edges

Edges connects two locations. Edges set the perimeter where something can move. Edges can be directed or undirected. If edge is undirected then it is modeled as <edge directed="false">

Edges contain the following fields:

- source – string, shows wherefrom the location directed
- target – string, shows where to the location directed
- directed – boolean [true, false]

Example (Fig.1):

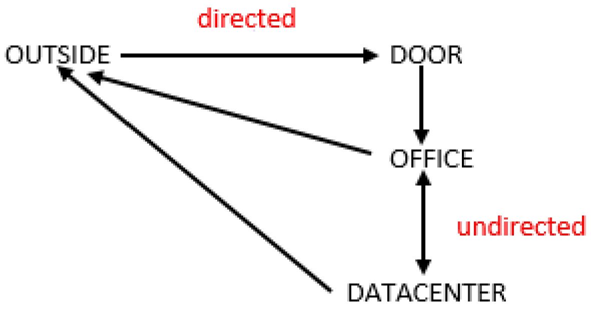


Figure 1. Edges in the TREsPASS Model

In the example in Figure 3, it is possible to move from the location outside to location office only through the door and go back outside without the need to pass any access control element (e.g. door). There are no access control element between the office and the datacenter and the actor can move back and forth freely. Therefore, it is undirected edge.

Actors

Actors can move along the edges that connect the physical locations. Actors may have a various roles. Roles set the privileges of the actors in the system – the set of the access credentials (keys) to pass through security control elements (e.g. door) and set of credentials to access certain types of information.

Actor contains the following fields:

- id – unique string, actor first name and last name
- atLocations – optional string

Example:

The actor by the name of the Finn (<actor id="Finn">) is located in the office (<atLocations>office</atLocations>).

Items

These are such elements, which cannot move on their own. They can have an owner and move with him. For example, an id card cannot move on its own.

Item contains the following fields:

- name – string
- id – string, unique
- atLocations – optional string, shows where the item is located

Example:

An item may be an idcard with id - x002, which is located at location Sydney.

Data

Data corresponds to any knowledge of an actor or data moving across communication networks. Data resides at locations. If the location is physical, it can be either an actor (Sydney knows the password) or an asset (security certificate residing on a chip card, in this case the chip card is an item).

Data contains the following fields:

- id – string, unique
- name - string
- value - string
- atLocations – string, optional

Example:

The employee’s Sydney (atLocations) password (data name) is “2345” (value).

Policies

Policies govern access to locations. A policy is always attached to a location.

It consists of two parts:

1. Credentials: items or data that the attacker must have or know to make an action.
2. Action. What is possible to make if the attacker has obtained all the required credentials.

Examples:

A policy attached to a location door might look like this: {key: move} (door)

It means that if an actor needs to move through the door he needs a key.

A policy attached to a location notebook might look like this: {password, idcard: In} (notebook)

It means that if an actor needs to execute data input action from a notebook he needs to know a password (data) and to have an ID-card (item).

Predicates

Predicates are auxiliary elements in the model. It is worth noticing that the role in the TREsPASS model is modeled as a predicate.
Predicate contains the following fields:

- id – model-wise unique string
- arity – a number, represented as a string
- value - string

Example:
Sydney has two roles: employee and administrator.

Processes

The processes govern data exchange flow between locations. Similarly to the policies the processes are attached to locations.

Example:

A location switch is likely to contain a process describing the packet routing protocol.

```
Policies = {[SW1] : {out(“IP”,,,,,,)}};
Processes = {
        in (“IP”, !dstAddr˜0.0.0.0/0, !dstPort, !srcAddrt, !srcPort, !request, !user)
        .out (“IP”, dstAddr, dstPort, srcAddr,srcPort, request, user) @SW1
        };
```

The process consists of two subprocesses: input and output, which are carried out sequentially. The „in” process is data input (Input), it filters data which the switch component accepts.
The example above shows us that we need the packets, where the type is “Ip”, the field "Adress" corresponds to this value: "!dstAddr ~ 0.0.0.0/0", which have the fields: port, the source address, the recipient's address, request and user. If the valid packet containing all the required data comes, the process “Output” starts and takes values, for example, the recipient's addresses and redirects this packet to the recipient's address and recipient’s port with the same value of request and user, taken from the packet, filtered by the "in" rule.

2.2 TREsPASS Workflow

The TREsPASS workflow consists of several steps:

1. A description of the target infrastructure (typically an SME).
2. Generating a navigator map.
3. Attack scenario generation.
4. The security analysis.
5. Result visualization.

Step 1: A description of the target infrastructure.

The TREsPASS workflow starts with the modeling effort where the users describe the system in any sort of graphical format, such as UML. Any editor can be used provided that it is capable of producing the XML-formatted navigator map, for example BiZZdesign Architect. [13] One of the goals of integration of TREsPASS into ISKE is that we can replace this step with ISKE assuming that the system has already been modeled in ISKE. Because in the public sector in Estonia many information systems have already been described as a model of ISKE. Then the navigator map can be generated from the ISKE.

Step 2: Generating a navigator map.

A navigator map (Fig.2) is a structural description of the target infrastructure in an XML format. The structure of the navigator map is described in Section 2.1.

In this step a navigator map is generated from UML model of the system description or from the ISKE model.

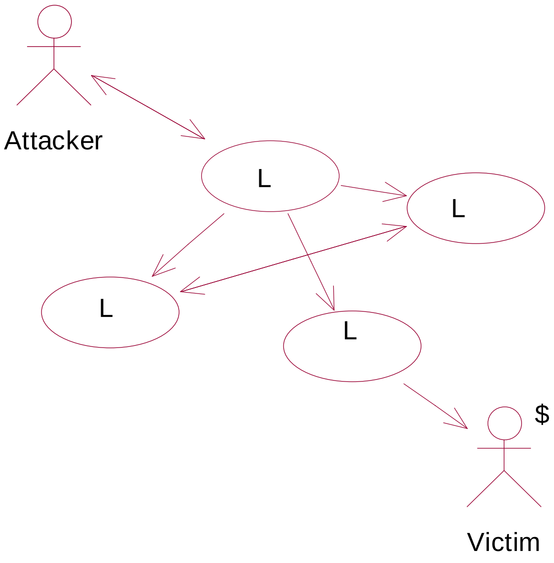


Figure 2. Navigator Map

Figure 2 shows an example of a navigator map, which contains of locations (denoted by “L”), an attacker and a victim. In this scenario victim has some money and the goal of the attacker is to get it.

Step 3: Attack scenario generation.

Attack scenario generation is done by the so-called attack generator, which relies on model checking and verification to check for the reachability of the undesired states of the system. Such on undesired state is an attacker goal.

The undesired state of the system against which we want to protect is fixed. In example, Fig.3 describes the state when the attacker has got victim’s money. If the undesired state cannot be reached attacks against such a system are infeasible. If the state can be reached all the paths, form the resulting attack scenario, which is encoded in the form of an attack tree (Fig.4).

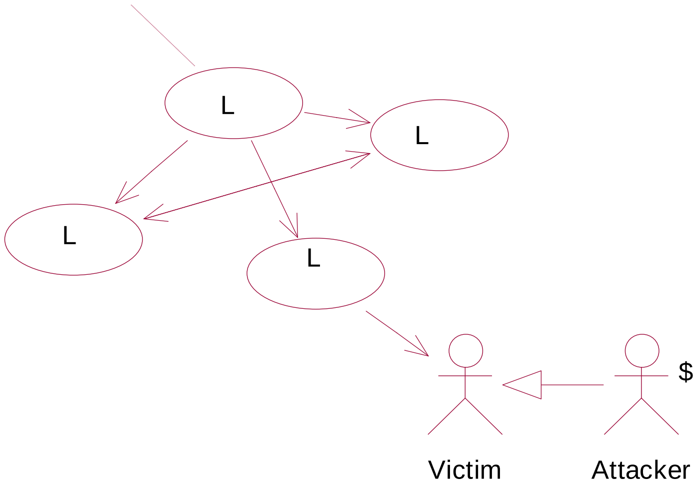


Figure 3. Model Checking and Verification

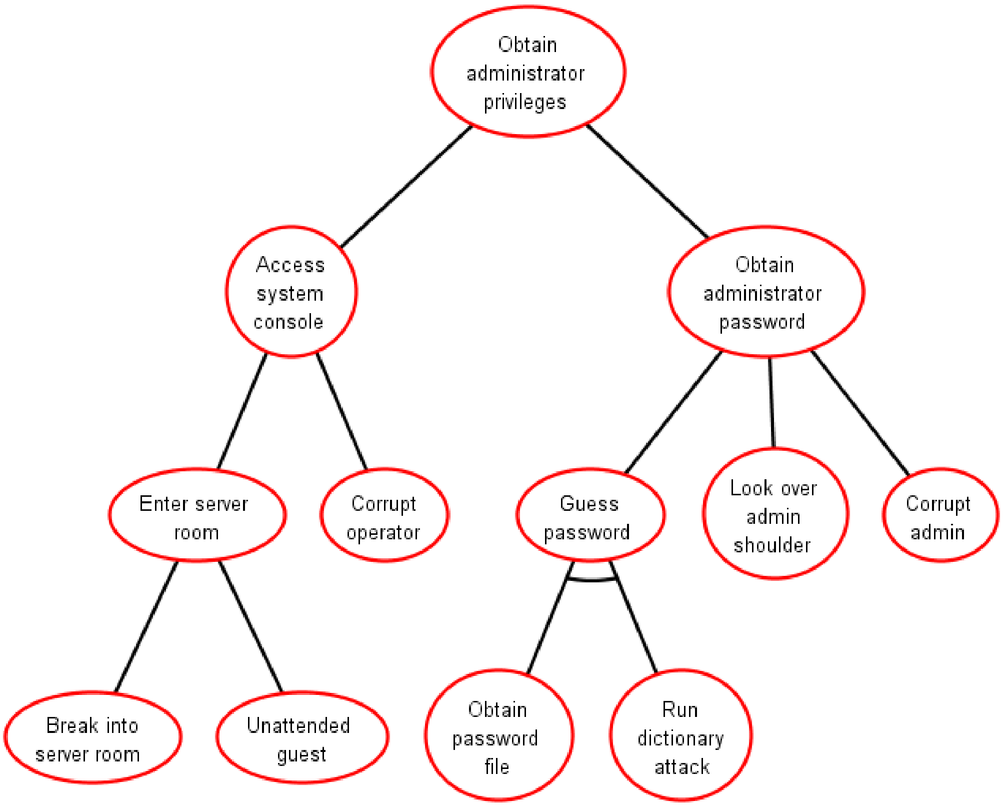


Figure 4. An example of an attack tree for attacking the server room [14].

Step 4: The security analysis.

Given an attack scenario, various analysis tools can be applied to analyze this attack scenario in terms of different parameters, for example:

- is it profitable to attack this system

- what security measures are better to use to protect from attacks
- what will be the expected losses (measured in EUR) from attacks
- when and where the most advantageous to attack
- what is the weakest point in the system.

Step 5: Result visualization.

The results are displayed to the end user and based on these results the system allows users to plan the protection of their company.

In the next section ISKE will be described.

3. ISKE

ISKE is a security standard that has been developed for the Estonian public sector. ISKE arrangement and development is based on a German BSI standard of information security - IT Baseline Protection Manual (IT-Grundschutz in German), which is suitable to the Estonian situation. [1]

3.1 ISKE goals and structure

ISKE goal is to protect and preserve assets like:

- data and databases;
- IT equipment;
- data sharing environments;
- Software.

ISKE consists of a general security arrangements and a description of the security measures. In addition to the technical measures, ISKE also contains recommendations for the organization, infrastructure and personnel.

ISKE is mandatory for the state and local government databases, information systems, but it can be used by commercial enterprises as well.

ISKE is a huge lookup table: typical modules for informational assets, associated threats and countermeasures. The security class of an asset determines the set of related countermeasures. In order to apply it, we list all information assets of an enterprise. Each information asset is assigned with a typical module and the desired security class is specified for each of the assets. Proceeding from this level, we get the list of required security measures.

To achieve the desirable security level it is required that each asset is assigned with the corresponding security class which consists of the three main security goals (CIA triad): information confidentiality, integrity, availability.

Based on the chosen security level the corresponding list of required countermeasures is produced.

Qualitative metrics to measure security level in ISKE:

- low
- average
- high

For example, K3T2K2, that designates availability - high, integrity – average, confidentiality – average.

Each security level corresponds to a certain ISKE security kit aiming at protecting assets.

The first version of the implementation guide ISKE was ready in October 2003. [9]

"*ISKE as baseline security system is one set of developed security measures, which will be applicable to all informational assets, regardless of their real security requirements. Contains more than 1,000 security measures.*" [1]

3.2 ISKE shortcomings

ISKE has some shortcomings. It is mandatory, not as a helpful handbook, designed for large corporations and hard for small and medium-sized enterprises. ISKE reference book consists of approximately 3000 pages. [1]

It has a long update cycle, but real-life threat landscape is highly dynamic. [15] [16]

ISKE has few typical modules while real systems are much more diverse. Even the IBM case study scenario [12] introduced components such as, for example the virtual machine, which is not present the in ISKE typical modules. Therefore from the point of view of ISKE it is unknown how to analyze these components. These new typical modules are added very seldom.

3.3 ISKE Tool

The ISKE tool [17] (Fig. 5) is helpful and supportive tool that allows to map information resources of an organization, to a set of security classes and corresponding security levels, to assign typical ISKE modules to information assets, to group and separate information assets, to create and manage the implementation of the plan, which will help to keep ISKE implementation process.

ISKE Tool is designed to reduce the time spent on security threats and directory processing and to allow implementers to follow the process of implementing in a semi-automated way.
It can be used for both local working environments as well as in the centrally configured databases.

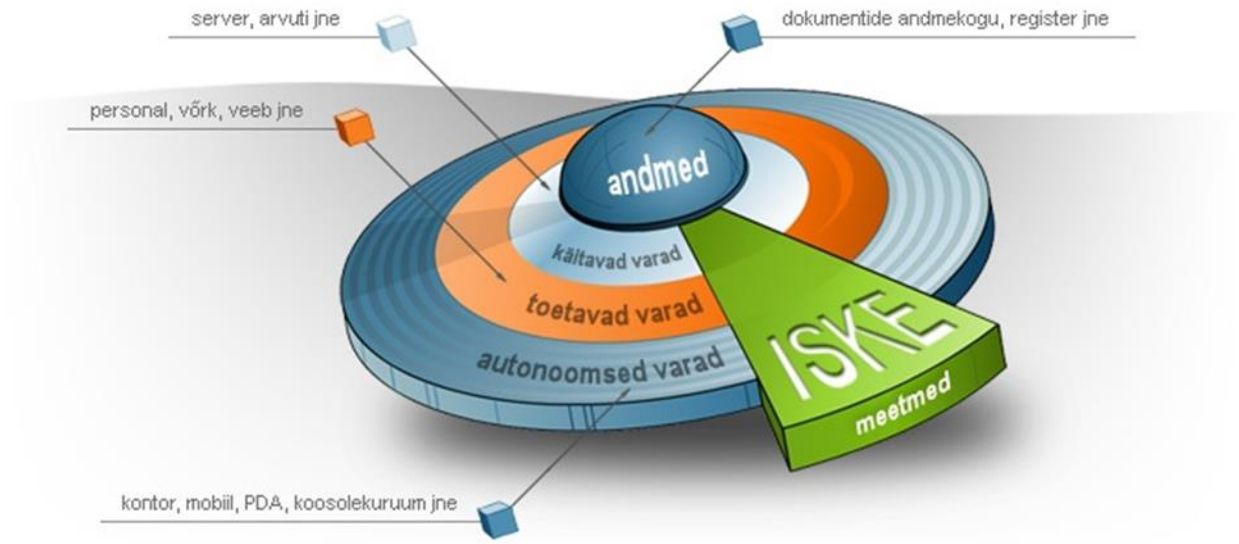


Figure 5. ISKE Tool overall view [17]

3.4 Comparison of ISKE and TREsPASS

Workflow similarity

In both cases workflow starts with the system description. Then there is a list of threats or the list of attacks and protection plan. In the TREsPASS Toolset a system is modelled at a higher level of granularity compared to ISKE. I refer the reader to Table 1 for the detailed description of the workflow in both of the cases.

ISKE Tool	TREsPASS Toolset
1. Describe the system	1. Describe the system
2. List of threats	2. Attack scenario
3. Countermeasures implementation plan	3. List of ranked countermeasures (Implementation plan).

Table 1. ISKE Tool and TREsPASS Toolset workflow

Differences in flexibility

ISKE is designed mainly for static systems. If we have described system once it is assumed that this description does not change in the short perspective. This standard is not designed to take into account rapidly changing dynamic environments and its update cycle takes considerable time. On the contrary, TREsPASS is capable of performing near real-time analysis keeping track of what is happening around (e.g. the activity of hackers). It updates automatically. Adding this dynamic flexibility to the ISKE Tool would be a fruitful goal.

Differences in analysis results

ISKE Tool generates unranked countermeasures implementation plan, but the TREsPASS Toolset generates prioritized (ranked) list of countermeasures. It indicates the weakest points in the system and shows which security controls it is most optimal to deploy at the moment.

4. Integration

As ISKE Tool generates only threats and protection measures and is unable to prioritize information security risks, but TREsPASS analyzes in near real-time and points which are the weaknesses in the system and acts as a decision support system highlighting which security measures are the most critical and should be implemented right now. I hope that by means of the TREsPASS Toolset the ISKE Tool will be improved.

The objective of integration is the possibility to analyze ISKE models with TREsPASS and TREsPASS models with ISKE.

It is desirable to analyze ISKE models with TREsPASS, because TREsPASS generates prioritized (ranked) list of countermeasures and shows which security measures are the most optimal to use at the moment and thus produces results containing more useful information.

Integration in the opposite direction is important as well, because ISKE Tool is mandatory in Estonia. In the beginning the system could be modeled in TREsPASS and then ISKE model could be generated out of it.

Currently we are missing a software tool which could import data from the TREsPASS Navigator Map and generate ISKE model out of it, equally as import data from the ISKE model and convert it into a TREsPASS Navigator Map.

To reach the above mentioned goals I have designed the central component called the integration governance component (Fig.6), governing all model transformation activities.

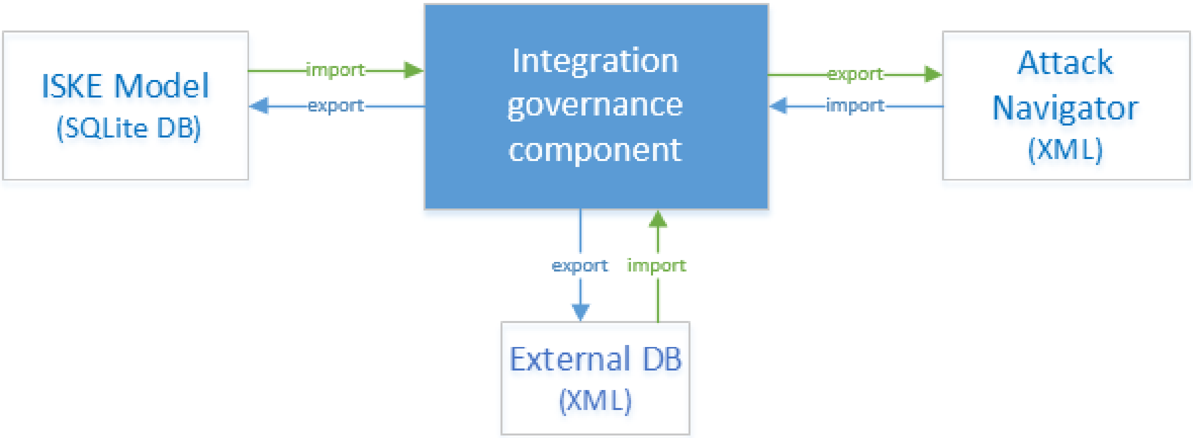


Figure 6. Integration scheme

To analyze the ISKE model with TREsPASS we import all the data, contained in the ISKE model. However, this set of data is only a subset of the information required to run the TREsPASS analysis, because the TREsPASS model contains much more data than the ISKE model does. It is hard to imagine policies and processes in the ISKE Tool, therefore an external database to store policies and processes is needed. Thus the data from the ISKE model and corresponding data from the external database get merged together and converted into the attack navigator format.

If we want to analyze the TREsPASS Model with the ISKE model, we load an XML file. All that makes sense in the context of ISKE, ends up there and everything else we write to an external database. This component may be any of: XML file, MySQL database, a text file.

4.1 TREsPASS Model (integration governance component)

The TREsPASS model component has to be self-sustained. This component itself has to contain enough information to make both types of integration.

Based on the data described in Section 2.1, I have created corresponding classes to represent the TREsPASS model data: locations, edges, assets, actors, policies, processes and predicates. Each of them contains a list of the relevant objects. For example, locations is a class that contains a list of objects of type Location, Edges – a class that contains a list of objects of type Edge.

The navigator map contains the following entities:

- **Location**

String id; // location is a string, it represented as ID, for example ID may be a laptop,
Domain domain; // Domain shows in what zone location is situated. I have a separate class Domain.java in which there is an enumeration: PHYSICAL, CYBER
String dislocation; // is optional string, description of location

- **Edge**

String source; // source is a string, it shows wherefrom the location directed
String target; // target is a string, it shows where to the location directed

boolean directed; // directed can be true or false. If directed = false, it means that location is undirected, then [u]: [location], if location directed, then [d]: [location]

- **Asset**

It is an abstract class. Data and Items extends this class.
String name; // string, shows data or item name
String id; // string, identifier must be unique
String dislocation; // is optional string, description of location

- **Actor**

String id; //string, actor's name and surname
String dislocation; // is optional string, description of an actor

- **Policy**

Policies and processes are stored as XML Node.

Node policy; //Root XML Node containing the subtree of policy nodes.

- **Process**

Node process; //Root XML Node containing the subtree of process nodes.

- **Predicate**

String id;
String arity; //As a predicate is key-value pair arity is always set to 2.
List <String> values;

4.2 Data transformation

Data transformation is achieved using the following functions: importFromXML(), importFromISKE(), exportToXML(), exportToISKE(). Model data can be imported/exported to/from ISKE database, as well as navigator map. An extra part not fitting into the ISKE model, such as policies and processes, end up in an external database.

If we transform ISKE model into navigator map, the model’s constructor receives a database file name, then it establishes a connection to the database (SQLite JDBC [8]) and the connection descriptor is passed subsequently to the corresponding constructors of objects: location, actor, etc. Each of this objects is smart enough to recreate itself given the connection handler by querying the database for the required data.

If we wish to transform the TREsPASS navigator map into an ISKE model we call a function importFromXML() (Fig.7) passing navigator map file into it. The function parses this XML file and passes subsequently DOM root note to the constructors of objects: location, actor, etc. Each of this objects is smart enough to recreate itself given the root node of the DOM tree. This process is illustrated in Fig. 7.

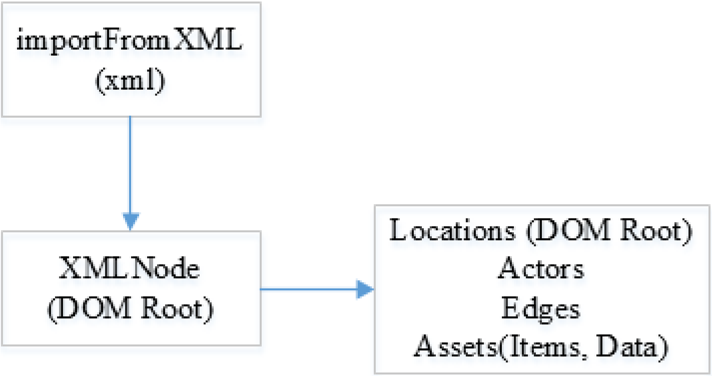


Figure 7. Import from navigator map.

Import/export to/from navigator map is done by means of XML DOM parser [7]. If we need to import data from a navigator map, the corresponding XML file is sent to the integration governance component, which in turn passes the XML ROM root element further to each of the subclasses and each class knows how to recreate itself from the file, it takes only the necessary information.

Importing data from the ISKE model and exporting into the ISKE model is done using SQLite JDBC. [8] If I need to export data into the ISKE model, the ISKE model database is prepared by SQL query “PrepareDatabase” which first removes corresponding zones, locations, actors, roles, and location groups from the database, then adds the two zones: cyber and physical, as well as creates asset groups: items, data and locations. Therefore, data from the TREsPASS model can be transferred into the ISKE model.

5. Modeling requirements

In order to be able to model something meaningful in such an integrated solution, some certain guidelines have to be followed:

- Guidelines for simulation – the two zones must be present: ‘physical’ and ‘cyber’. It is necessary to model an enterprise in terms of locations. All information assets must be locations.
- Knowledge base - it can be empty, analysts can create their own components, so I assume that all the necessary components are already created, they are already there, and I will take them. Knowledge base can be realized in the form of XML, MySQL, etc. In my work, I decided to use XML format.

Before starting modeling, ensure that:

- Zones named 'physical' and 'cyber' exist.
- Asset groups named 'locations', 'items', and 'data' exist.

Creating an ISKE model:

Treat an enterprise infrastructure as a set of interconnected locations with actors, data, and assets moving around along connections between the locations.

A list of requirements for the corresponding model entities can be observed below.

Locations

- Ensure that id is set.
- Ensure that asset group is set to 'locations’
- Ensure that zone is set,
- Dislocation is optional, it can be empty.

Edges

- Place information on edges in the description field of locations.
- Edges may be directed (d: [location]) or undirected (u:[location]).
- Connections are space separated.

Items

- Ensure that name and id are set.
- Ensure that asset group is set to ’items’.
- Ensure that zone is set to ’physical’.

Data

- Ensure that name and id are set.
- Ensure that asset group is set to ’data’.
- Ensure that zone is set to ’cyber’.
- Ensure that value is set.

Actors

- Treat name or surname as id (must be unique). It would be better to use their combination as id.
- Treat description as dislocation (optional field).

Predicates

- Currently all are exported to external database.
- For future development it would make sense to keep roles in the ISKE model.

6. Results

As a result of my work I have created a program that enhances the ISKE Tool with analytical capabilities of the TREsPASS Toolset, that is capable of converting TREsPASS models into ISKE models and vice versa.

The user interface of the tool consists of two tabs and the user can select the desired action: transform TREsPASS model into ISKE model (Fig.8) or transform ISKE model into the TREsPASS model (Fig.12).

- **Export to ISKE (Fig. 8)**

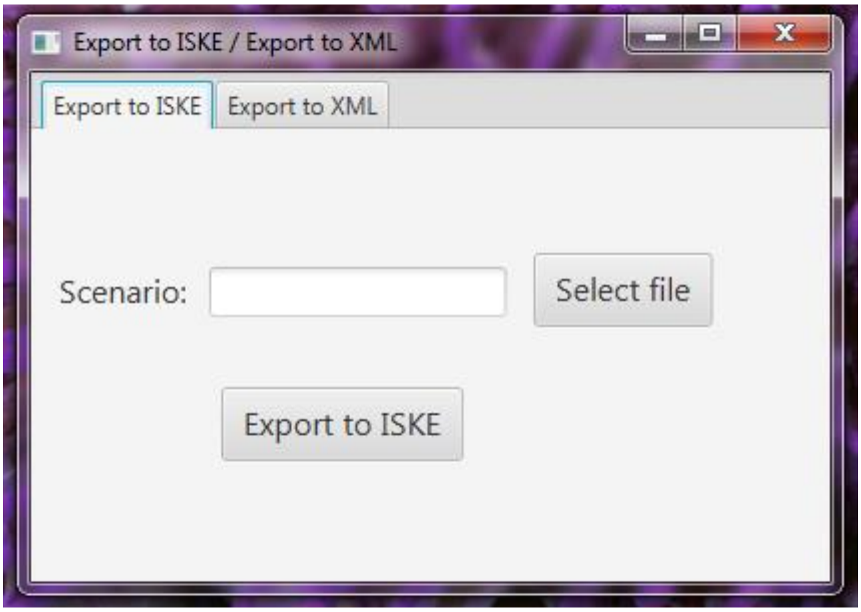
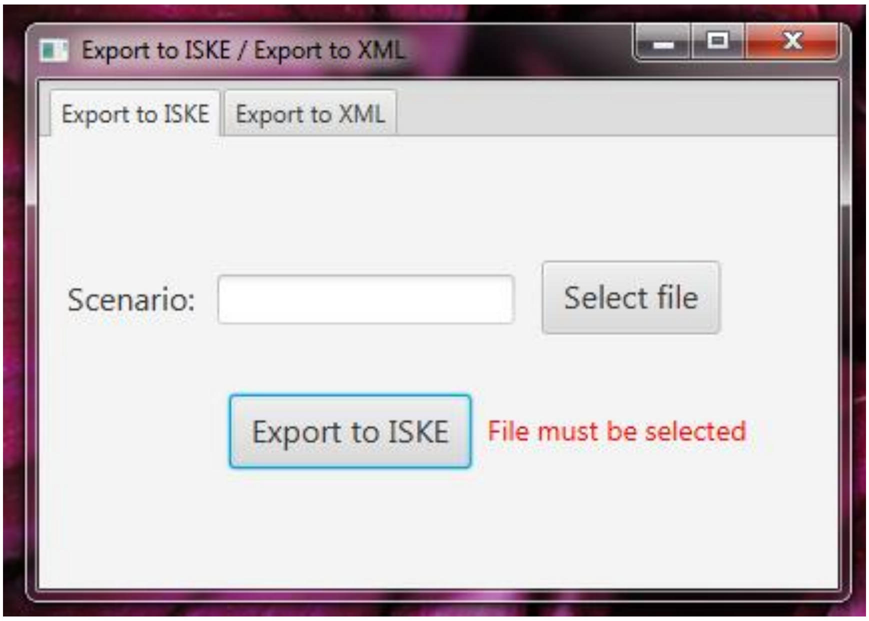


Figure 8. Main window of Export to ISKE

First the user must select the TREsPASS model to convert into the ISKE model. If the TREsPASS model is not selected and the user has pressed the button “Export to ISKE”, the program returns a message “File must be selected” (Fig.9):

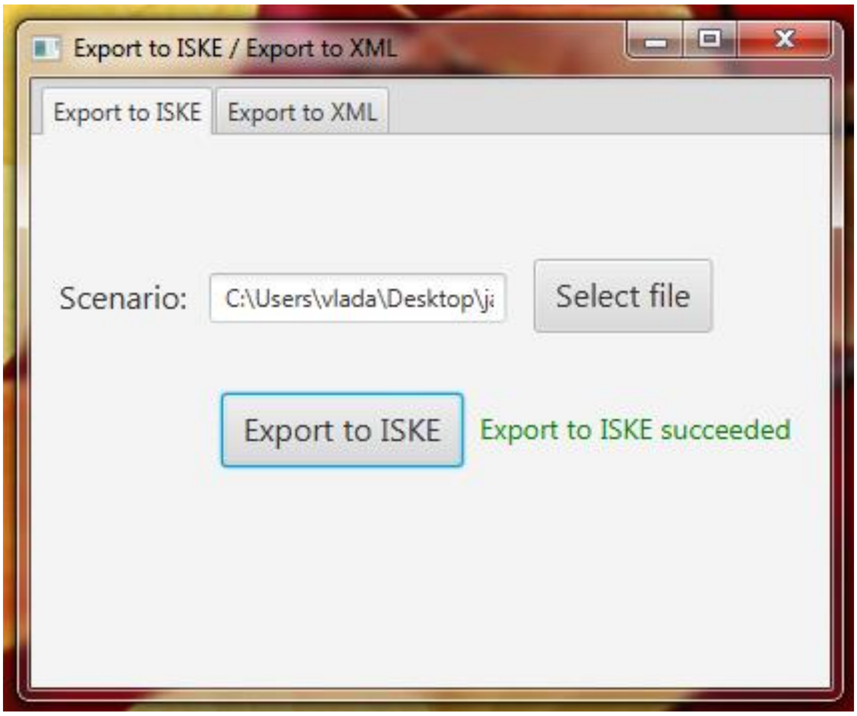


Scenario:

File must be selected

Figure 9. An error of Export to ISKE

After the user has selected the TREsPASS model and pressed the button “Export to ISKE” he will be presented with two dialog windows “Save database” where the user can select the ISKE model database and “Save external database” and select the file for the external database. If the export was successful, the message “Export to ISKE succeeded” appears (Fig.10).



Scenario: C:\Users\vlada\Desktop\ja

Export to ISKE succeeded

Figure 10. Successful export to ISKE

Fig. 11 shows the step by step process of conversion of the TREsPASS model into the ISKE model:

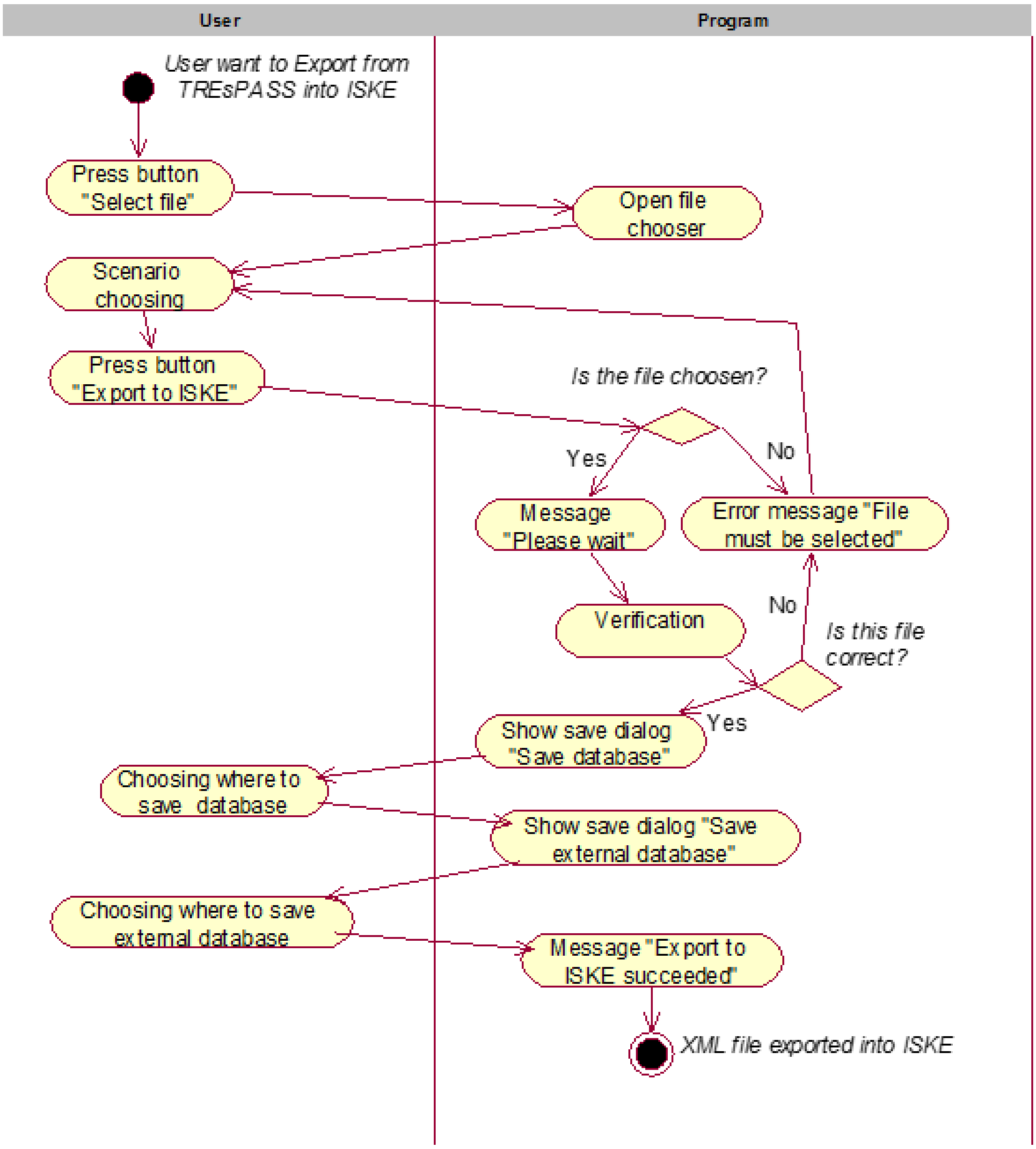

User
Program
User want to Export from TREsPASS into ISKE
Press button "Select file"
Open file chooser
Scenario choosing
Press button "Export to ISKE"
Is the file choosen?
Yes
No
Message "Please wait"
Error message "File must be selected"
Verification
No
Is this file correct?
Yes
Show save dialog "Save database"
Choosing where to save database
Show save dialog "Save external database"
Choosing where to save external database
Message "Export to ISKE succeeded"
XML file exported into ISKE

Figure 11. Export to ISKE Workflow Diagram

The left column shows the user's actions and right the program's response to the user's actions. For example, if the user presses the button „Export to ISKE", the programm checks if the file has been selected or not. If the file was selected, there is a verification step and if the file was correct the programm shows the message"Save database". Otherwise the program displays an error message „File must be selected" and user must select the file again.

- **Export to XML (Fig.12):**

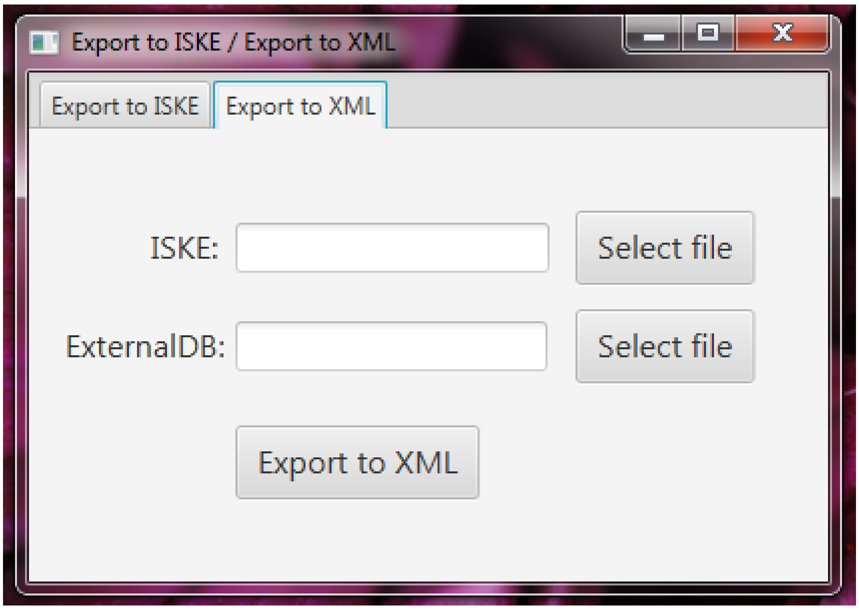
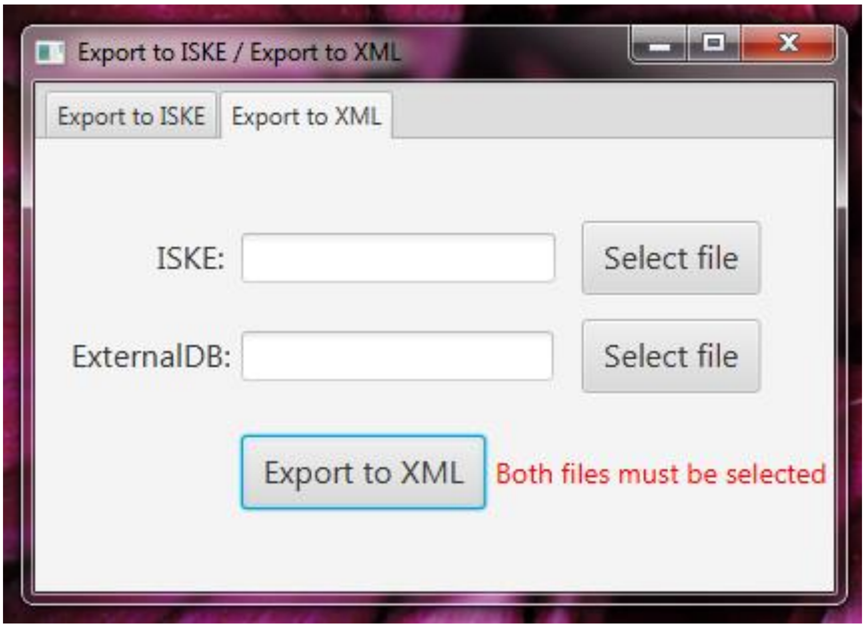


Figure 12. Main window of export to XML

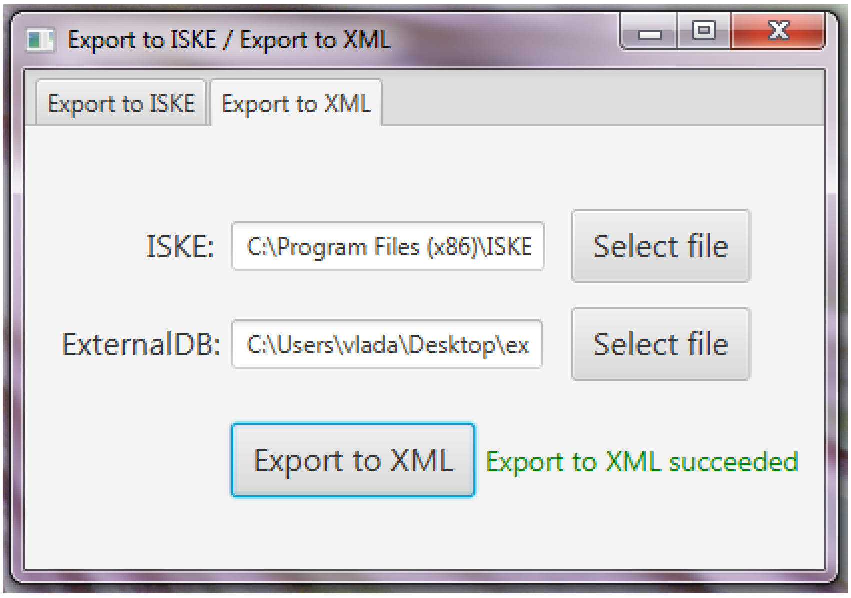
To do a conversion in the opposite direction from ISKE to TREsPASS the user must select the ISKE database to import as well as external database with required external components. If at least one of them is not selected and pressing the button “Export to XML” would result in the following error message (Fig.13):



Both files must be selected

Figure 13. An error of export to XML

If the ISKE database and the external database are selected, and the button "Export to ISKE" has been pressed a dialog window "Save XML file" will be opened. The user must provide a name to the file, under which it will saved and must choose where the new file shall be created. If the export was successful, the message "Export to XML succeeded" appears (Fig.14).



ISKE: C:\Program Files (x86)\ISKE
ExternalDB: C:\Users\vlada\Desktop\ex
Export to XML succeeded

Figure 14. Successful export to XML

Fig.15 shows the process converting ISKE model into the TREsPASS model:

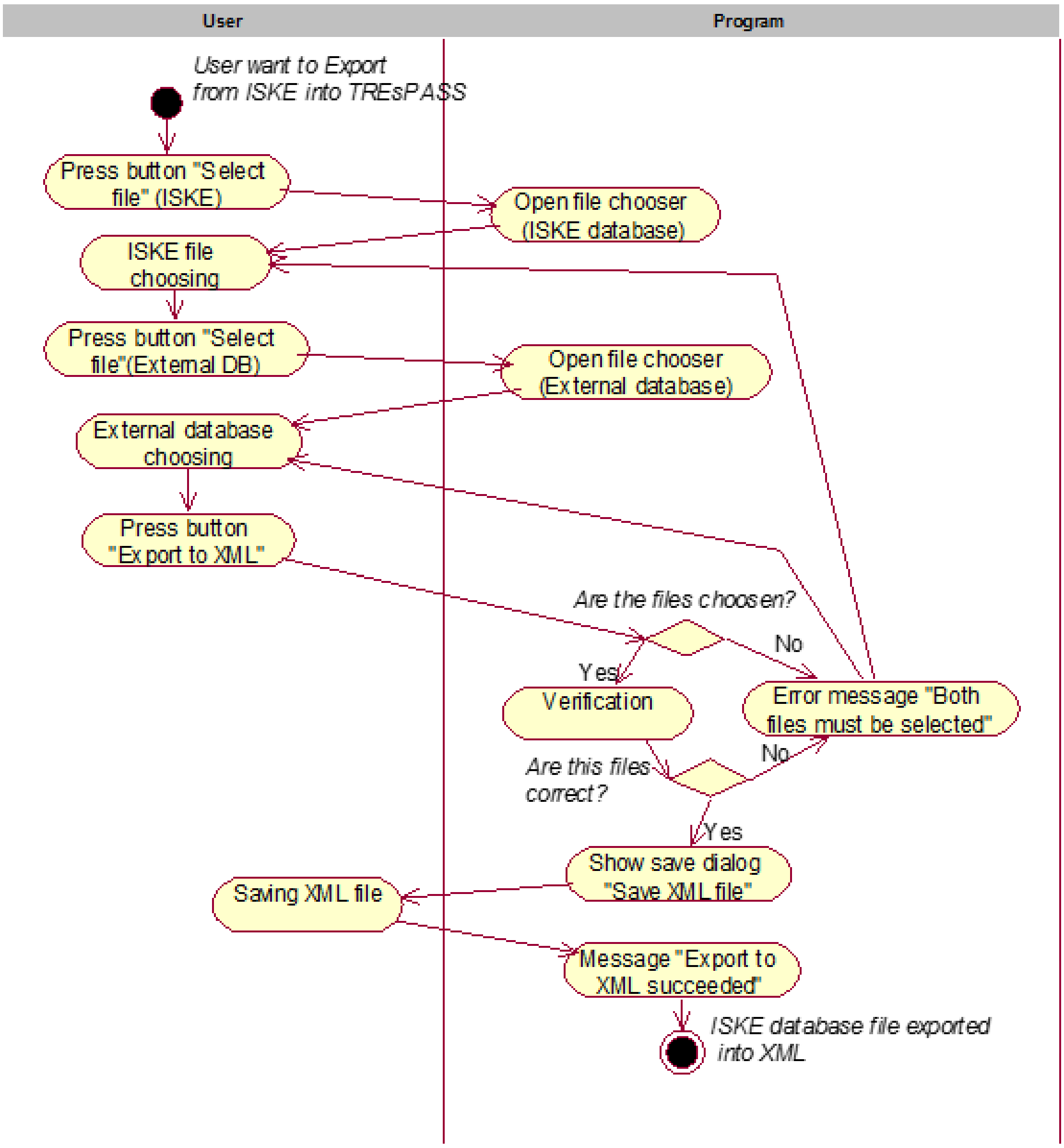

User
Program
User want to Export from ISKE into TREsPASS
Press button "Select file" (ISKE)
Open file chooser (ISKE database)
ISKE file choosing
Press button "Select file"(External DB)
Open file chooser (External database)
External database choosing
Press button "Export to XML"
Are the files choosen?
Yes
No
Verification
Error message "Both files must be selected"
Are this files correct?
No
Yes
Show save dialog "Save XML file"
Saving XML file
Message "Export to XML succeeded"
ISKE database file exported into XML

Figure 15. Export to XML Workflow Diagram

The left column shows the user’s actions and the right column - the program’s response to these actions. For example, if the user presses the button „Export to XML“, the programm checks if the file has been selected or not. If the file was selected, there is a verification step and if file was correct the programm shows the message“Save XML file“. Otherwise the program displays an error message „Both files must be selected“ and the user must select the file again.

7. Kokkuvõte

Käesoleva töö põhieesmärk oli luua program, mis täiustab ISKE rakendustööriista analüüsivõimet TREsPASS rakendustööriistade abil, realiseerides seda funktsionaalsust iseseisvas tarkvaras. Samuti pakkusin modelleerimise nõuete nimekirja ISKE rakendustööriistale.

Selle probleemi lahendamiseks mina uurisin ISKE ja TREsPASS rakendustööriistu, kasutades IBM näidisstsenaarium [12] mis oli kasutatud TREsPASS projektis, modelleerisin seda stsenaariumi ISKE rakendustööriistas ja võrdlesin andmetüüpe. Samuti lõin klassid TREsPASS andmete hoidmiseks, funktsioonid nende andmete salvestamiseks ISKE rakendustööriistasse, ISKE rakendustööriistast allalaadimiseks ja XML formaadis salvestamiseks.

Minu töö tulemuses olen loonud programm, mis integreerib ISKE mudel TREsPASS mudelisse (XML failisse) ja vastupidi (vaata 6. peatükk). Programm on kirjutatud Java programmeerimiskeeles.

Me soovime kasutada TREsPASS mudelid ISKE rakendustööriistas ja analüüsida neid turbekataloogi seisukohalt. Samuti on palju ISKE mudeleid ja me tahame integreerida neid TREsPASS süsteemi ja analüüsida neid kogu funktsionaalsust kasutades. Programmi koostamisel leidsin, et ümberkujundamist ei saa teha sirgjooneliselt, sest on palju nüansse. Selle jaoks peab ISKE modelleerimise protsess vastama teatud nõuetele. Sellepärast pakkusin modelleerimise nõuete kogum ISKE rakendustööriistale (vaata 5. peatükk). Tuleviku töö lihtsustamiseks pakkusin ISKE edasiarendamisele nõuete nimekirja (vaata Future plans).

Summary

The aim of the thesis is to create a program that enhances the ISKE Tool with analytical capabilities of the TREsPASS Toolset, implementing this functionality in the form of stand-alone piece of software. Additionally a list of modeling requirements for the ISKE Tool was suggested.

To reach the objectives I learned the TREsPASS Toolset and the ISKE Tool, taking an example TREsPASS project-internal case study by IBM [12], modelled it in the ISKE Tool and compared the data types. Classes for storing the TREsPASS model data have been created, as well as corresponding functions for storing this data in the form of ISKE model, uploading from ISKE Tool and storing back to XML.

As a result of my work I created a component that integrates ISKE model into the TREsPASS model (to XML file) and the opposite way (see Section 6). The component is implemented in Java programming language.

We want to use TREsPASS models in ISKE and analyze them from the viewpoint of security catalog. A lot of ISKE models exist and we want to integrate them into the TREsPASS analysis toolchain and analyze them with all functionality. I discovered that transformation cannot be done as is due to the existence of numerous subtleties. In order to do that the ISKE modeling workflow must conform to a set of certain requirements. Therefore I suggested these (see Section 5). To facilitate work in the future, I offered a list of further development requirements for the ISKE Tool (see Future plans).

Future plans

To make it easier to work with the ISKE Tool, I suggested some modifications to the ISKE Tool. The list of suggested modifications is the following:

1. Re-design connections between assets, such that a connection may be directed or undirected.
2. Extend the notion of position to actors as well.
3. Extend Class B asset types to include such entries like items, data, and predicates.
4. Add special asset categories, such as location and access control element (e.g. door). In this case such a notion of access control element would be location itself.
5. Enhance the notion of location with ability to attach an access control policy to it.
6. Introduce the notion of a predicates (e.g. trust).
7. Introduce the notion for process.

To facilitate even more tight integration of TREsPASS and ISKE the further implementation for the ISKE Tool will be focused on heuristics mapping typical models from ISKE to various entities in the TREsPASS model as well as deriving meaningful security class annotations on the assets in an automated way.

Further development will focus on converting the TREsPASS model into the ISKE model, as it makes sense to analyze the system with the TREsPASS toolset first and then add more details to analysis by generating the ISKE model out of the TREsPASS model and analyzing it in ISKE.

List of references

[1] Presentation on ISKE [WWW] https://www.ria.ee/public/ISKE/ISKE_english_2012.pdf (23.02.2015)

[2] New TREsPASS flyer [WWW] http://www.trespass-project.eu/sites/default/files/Deliverables/flyer_2015-01-d.pdf (23.04.2015)

[3] XML Tutorial [WWW] http://www.w3schools.com/xml/ (6.05. 2015)

[4] FP7 [WWW] http://ec.europa.eu/research/fp7/understanding/fp7inbrief/what-is_en.html (5.05.2015)

[5] SQL Tutorial [WWW] http://www.w3schools.com/sql/ (6.05.2015)

[6] The Unified Modeling Language [WWW] http://www.sparxsystems.com.au/platforms/uml.html (6.05.2015)

[7] The Java Tutorials [WWW] https://docs.oracle.com/javase/tutorial/jaxp/dom/readingXML.html (6.04.2015)

[8] SQLite JDBC Driver [WWW] https://bitbucket.org/xerial/sqlite-jdbc (6.04.2015)

[9] Riigi infosüsteemi teejuht [WWW] https://www.ria.ee/teejuht/riigi-infosusteemi-olemus-ja-komponendid/infosusteemide-turvameetmete-susteem-iske (29.04.2015)

[10] Lenta.ru uudised [WWW] http://lenta.ru/news/2014/10/03/jpmorgan/ (2.05.2015)

[11] J. Plehhanova, „Assessment of integration possibilities of the TREsPASS toolset into the ISKE Tool," Tallinn, 2014.

[12] C. W. Probst, „TREsPASS cloud infrastructure model," 2014.

[13] BiZZdesign [WWW] http://www.bizzdesign.com/tools/bizzdesign-architect/ (28.04. 2015)

[14] J. Weiss, „Attack tree" *A system security engineering process. In Proceedings of the 14th National Computer Security Conference*, pp. 572-581, 1991.

[15] Pieters, W., Hadziosmanovic, D., Lenin, A., Montoya, L., Willemson, J.:Trespass: Plug-and-play attacker proles for security risk analysis (poster). In:35th IEEE Symposium on Security and Privacy, Los Alamitos, CA, USA, IEEE Computer Society, 2014.

[16] Lenin, A., Willemson, J., Sari, D.P.: Attacker proling in quantitative security assessment based on attack trees. In Bernsmed, K., Fischer-Hübner, S., eds.: Secure IT Systems - 19th Nordic Conference, NordSec 2014, Troms, Norway, 2014.

[17] Elektrooniline Riigi Infosüsteemi Amet - ISKE rakendustööriist [WWW] https://www.ria.ee/isketooriist/ (18.02.2015)