Attack-Defense-Tree Driven Creation of Protection Profiles

Master en management de la sécurité des systèmes d'information
2012-2013

Professional Project Final Report

Tejeddine Mouelhi
SnT
University of Luxembourg

Academic supervisor: Dr. Djamel Khadraoui
Local supervisor: Dr. Jean Schweitzer

Acknowledgement

I am thankful to my advisors, Dr. Djamel Khadraoui and Dr. Jean Schweitzer and to all those who helped me finish this project.
I also would like to thank my colleagues from the SnT, especially Dr. Barbara Kordy and Patrick Schweitzer who helped me in improving the project and provided guidance and advices on how to best use the ADTrees.

Résumé

Ce projet professionnel de Master a été effectué dans le cadre du projet ESA DG-Trac (Suivi et Traçage du transport de marchandises dangereuses dans le domaine médical). L'objectif a été de proposer une nouvelle approche pour créer les PP (profils de protection). Les profils de protection sont basés sur la méthodologie des critères communs (le standard ISO/IEC 15408). Nous avons proposé dans ce projet une nouvelle approche qui vise à offrir un processus bien structuré pour automatiser la génération des profils de protection. Cette approche se base sur la définition d'un arbre attaque-défense (ADTree). A partir de ce modèle les attaques et les défenses du système sont explicitement listées. Nous avons proposé un processus qui prend en entrée un ADTree. L'utilisateur doit ensuite compléter avec des informations additionnelles ce modèle avant de générer une version initiale du document contenant le profil de protection. Nous avons appliqué notre approche avec succès sur le cas du système DG-Trac « Traçage et suivi de marchandises dangereuses ». C'est un projet ESA dans lequel le SnT est impliqué. Nous avons pu montré ainsi via un exemple concret de l'approche peut être appliqué et est utile en pratique. Cette approche a été implantée dans un outil appelé ADTool qui a été développé auparavant par une équipe de recherche de l'université du Luxembourg.

Summary

In this work, we propose a new approach that aims at automating the generation of the protection profile documents from an attack-defense tree (ADTree). The idea is to enable the user to create an ADTree that represents the list of threats that a system faces, in addition to the defenses put in place to mitigate those threat. Based on this model, we are able to automate the generation of the protection profile document. The user still needs to enter some information to complete the generated document. This work was done in the context of the DG-Trac project, an ESA funded project that aims at building « Dangerous goods tracking and tracing » system. We have applied our approach to create the protection profile for DG-Trac. We have also implemented the approach in the ADTool, which is a modeling tool developed by a research group in the university of Luxembourg. This tool is open-source and is publically available for download.

Déclaration d'honnêteté

Je confirme par la présente que j'ai réalisé ce mémoire de Master sans aucune aide illicite d'autrui. Ce travail a été réalisé en ce conformant aux règles d'honnêteté intellectuelles.

Tejeddine Mouelhi

Table of Contents

Table of Figures

1 Introduction

Security is becoming more and more important nowadays. Several companies are considering this aspect when buying new IT Systems or when developing new products. It is therefore important to be able to guarantee that IT products are secure for both vendors and buyers. There should be a way to guarantee a minimum level of security in a standard way. The ISO/IEC 15408 was proposed to this end, in order to provide a common framework for both vendors and sellers to evaluate IT Systems from the point of view of security.
The process relies on defining a security target (or a protection profile), which includes a set of security functional requirements (for securing the system) and assurance requirements (to provide confidence in the implementation/quality of the security mechanisms).
The protection profile (PP) is a generic security target and allows providing for a class of IT Systems the set of security functional requirements and security assurance requirements in a generic way, independently from the technology and the specificities of the system to be evaluated.
The creation of a protection profile is therefore a critical task. In fact, a protection profile poorly defined does not provide any guarantee that the evaluated products are secure. It is therefore important to be able to define high-quality protection profiles that enable to tackle all the security threats and makes the product secure. What is needed to reach this objective is a clear methodology that guides the creation of the protection profile.
This project aims at tackling the issue of generation of protection profile by providing a new methodology relying on *attack-defense tree* (ADTree). ADTree is a graphical modeling tool that enables to represent the attack scenarios is a form of a tree. The main attack goal can be refined and decomposed into sub-goals. Along with the attacks, the ADTree represents the defenses mechanisms to protect against the threats. We rely on this modeling tool to provide a new process that supports the creation of protection profiles.
The proposed methodology does not provide a full automation of the creation of the protection document. This approach aims at providing a basic methodology for guiding the creation of the protection profile by using a good and easy-to-use graphical tool, the attack-defense trees.
This project was done in the context of an ESA funded project called DG-Trac (Dangerous Good Tracking and Tracing in the Medical Sector). This project involves several industrial partners including EPT, HITEC (a company developing and providing engineering and software solutions) and T&E (A German transportation company), in addition to the research institution the CRP Tudor and finally the SnT, research center. It is feasibility study project that aims at studying the opportunity to develop a DG-Trac solution. In this project, we are involved in the security requirement definition. Our objective is to define a protection profile for DG-Trac.
The approach proposed in this project will be applied to the DG-Trac project. We will try to show that the approach is feasible in practice and could be applied to a real example.

The remainder of this report is organized as follows. The first chapter presents the context of this work and provides the main notions needed to understand the contribution. The second chapter presents an overview of the state of the art. It will

show previous work related to the area of protection profile and techniques to improve the PP creation process, and will also present the more general related work in the area of security requirements elicitation, which can be considered close to the area of creation of PP. The third chapter presents in detail the approach and describes step-by-step, how the protection profile is created from the attack defense tree. It will also show a concrete example of creation of protection profile for the DG-Trac system.

2 Background and state of the art

This section introduces the context by presenting the ESA project DG-Trac, in which this work has been done. It will also present the concepts needed to understand the work done during this master project. It will present the common criteria methodology and introduce the protection profile concept. Afterward, this section will provide an overview of the state of the art, beginning with directly related work on approaches aiming at improving the process of creating protection profiles. Then, it will widen the scope and present the approaches related to more general aspects concerning security requirements elicitation in order to give an overview of existing work related to the methodologies proposed to define the list of threats a system faces and the security requirements needed to tackle them.

2.1 The DG-Trac project

This description of the DG-Trac is taken mainly from the project deliverables and the project proposal documents.

The DG-Trac project focuses on the tracking and tracing of dangerous goods' transport in the "specialized" medical sector. The number of dangerous goods transported increases every year, therefore so does the potential for incidents, ranging from major traffic accidents to theft and loss. The effects of these incidents can quickly become a matter for public concern. The tracking and tracing of dangerous goods' transportation throughout its journey would allow the various stakeholders in the process to react to alerts and to make relevant decisions based on real time information.

The customer(s), who orders the transport of these dangerous goods, as well as the logistic company carrying out the transport, needs to know where those goods are, and/or where they may pass, at any time as well as whether they have arrived safely at the destination. Additionally there is a need for public safety services to be aware of these goods and to know how to handle them correctly in case of any potential or actual incident.

Within this general transport area, medical transportation is a special case and is very often underestimated, as are its unique risks. The medical transport sector represents an area with very particular features and specific needs, some of which are detailed in the following list:

Medical goods are often sent in small quantities and may be consigned together with other goods that are not dangerous, leading to ambiguity in the overall package status.

Some transported medical goods are dangerous in even the smallest quantities, e.g. infectious substances such as the Ebola virus, Hepatitis B virus or the HIV virus and these can provoke a catastrophe if they are not controlled carefully.

There are stringent requirements concerning the security and privacy of the associated information as medical transport also includes matter (e.g. samples, blood and organs) belonging to individual citizens.

A key problem today is that in normal situations, neither the local, national nor European civil authorities have an overview or situational knowledge of particular dangerous goods transport.

A consequence of this lack of information is that in case of an alert and/or emergency, civil security authorities have no knowledge of the nature and characteristics of the goods transported and are therefore unable to take risk-reducing preventive measures.

This results in potentially dangerous situations, or causes delays and/or inappropriate reactions to events, with human, economic and environmental consequences.
Currently, there are no solutions available on the market today, and no known project planned that deals with the specifics of the transport and tracking of dangerous medical goods.
The DG-Trac project intends to address the above special problems in tracking and tracing dangerous medical goods and to introduce a centralized mechanism for the management and (controlled) distribution of information concerning each dangerous goods transport in Luxembourg, (called DG-Trac-Service).
In case of any incident, the emergency services can use this service to check if a transport containing dangerous goods is involved in the incident (or if one is near to the accident location). In this manner the first responders can be warned to take care before approaching any vehicle or material at the accident site.
To make this service attractive for logistic companies and end users the DG-Trac service will also provide additional services for them such as monitoring functions for transport quality (e.g. temperature and humidity) and easily consolidated access to the transport data of different logistic companies.
The project consortium intends to start with the scenario of dangerous goods transport in the specialized medical sector in Luxembourg because this is a little known but thriving area of dangerous goods transporting, and one where there is a strong demand for technology to make tracking and tracing possible.
The first step of this project will focus on the feasibility analysis. This feasibility study will investigate a potential integrated solution and its associated services that support the transport logistics for medical appliances, samples and waste (i.e. blood, surgical instruments, organs, etc.).
The objectives of the feasibility study are to collect requirements from all stakeholders for a DG- Trac service and analyze the state of the art technologies. Based on the results of this analysis, a system and service design of the DG-Trac service will be defined that allows fulfilling most of the stakeholder requirements based on the available technology. In a viability study, the financial and non-financial viability of the design and the proposed service will be evaluated. A proof of concept will show the principle implementation of such a service and demonstrate the technical feasibility. Based on the results of the viability analysis and the proof of concept, a roadmap will be defined how a sustainable service for the tracking and tracing of dangerous medical goods in Luxembourg can be established.
If the results of the feasibility study are positive and show a financial potential for this service it is the intention of the consortium to lunch a follow on demonstration project.

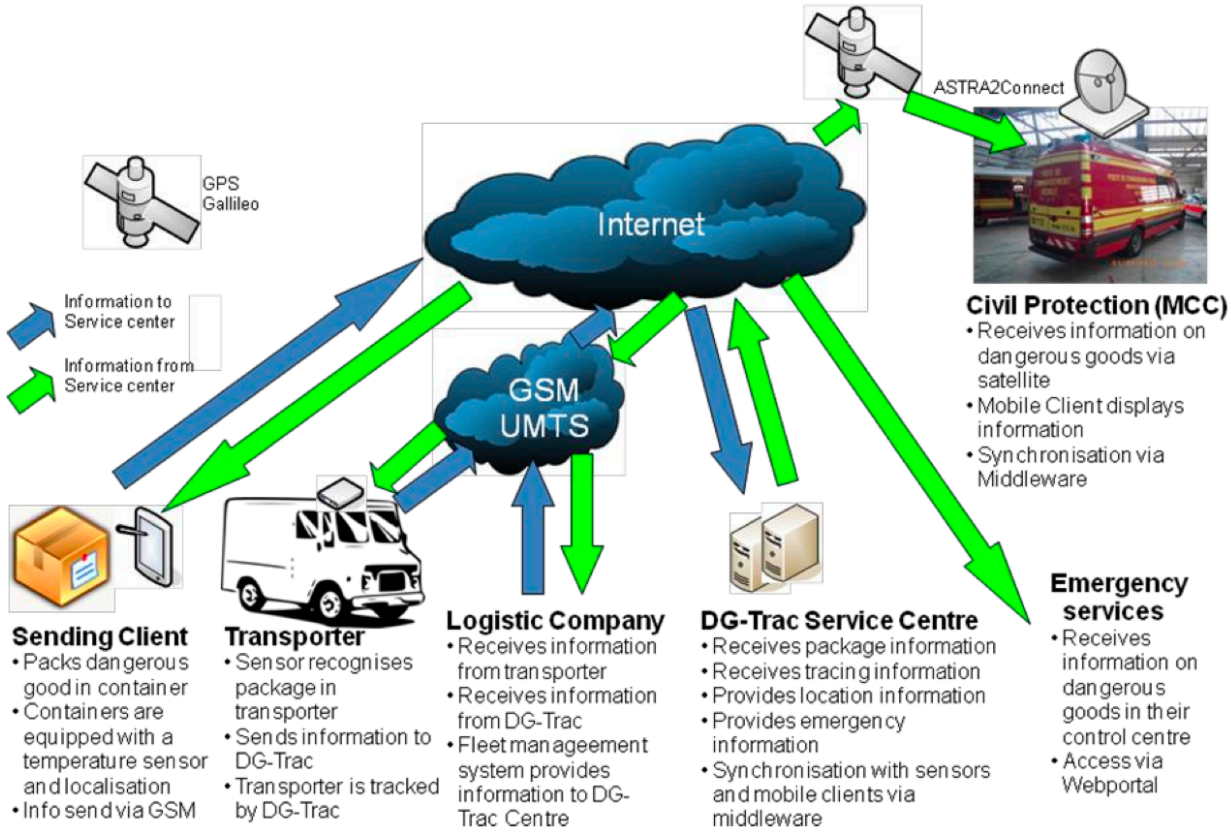

Figure 1 - The DG-Trac solution architecture

The main characteristic of the DG-Trac solution is that it will leverage existing networks and solutions as much as possible. This has the added advantage of allowing users to profit from their already existing equipment and thereby reducing the need to invest large sums of money in a completely new infrastructure.

The architecture of the service is shown in Figure 1. The sender and the receiver are connected to the DG-Trac Service via a standard Internet connection. Which devices will and can be used is dependent on the user network implementation. The DG-Trac solution will be independent of these decisions.

If active localization of containers is needed at the sending/receivers site, standard satellite navigation receivers (GPS, later Galileo) and/or off the shelf interior localization solutions (i.e. Wi-Fi localization) will be used.

The communication of the transporter with the DG-Trac Service Centre will be done via standard GSM/UMTS networks, as these networks are available over practically all of Europe. The focus o this project will be on the DG-Trac Service Center system that will be in charge of collecting information and sharing it with the customers and the emergency service (if needed).

2.2 Background on common criteria and protection profiles

Common Criteria (CC) [1] is a methodology for evaluating the security of IT systems. The methodology relies on ISO/IEC 15408 standard. The last version was published on July 2009 (Version 3.1).
CC methodology relies on defining a security target, which will represent the security requirements that the system should meet to pass the evaluation and reach the required assurance level.
In the document called "CC part 2", it offers a full catalogue of security requirements covering all aspects of security functional requirements and security assurance requirements.

Protection profiles represent the generic version of security target. They are independent of the underlying technology and enable to represent the set of security functional and assurance requirements for a given class or category of IT products.
According to ISO/IEC TR 15448, a protection profile should contain the following section:

- A description of the target of evaluation (TOE), the system to be evaluated.
- A description of the security environment that involves defining the threats targeting the system and its environment in addition to the assumptions and the organizational security policy.
- To cover the security threats a set of objective should be provided.
- To fulfill the security objectives a set of requirements are provided taken from the CC catalogue of security functional requirements and security assurance requirements (from CC part 2 and CC part 3 respectively).
- A rationale that explains how the requirements fulfill the objectives and how the latter cover the identified threats.

There are several protection profiles publicly available. For instance, several protection profiles are available in the USA National Information Assurance Partnership a program [2] initiated jointly by the NSA (National Security Agency) and by NIST (the national institute of standards and technology). There are for instance Antivirus Protection Profile, Operating System Protection Profile and Virtual Private Network (VPN).
Protection profiles can be used as a reference for a given class of products on the security requirements. According to the standard ISO/IEC 15408 [3-5]: "PP is used as a baseline defined by a group of IT developers, who then agree that all IT that they produce of this type will meet this baseline."
In this project we are using the protection profile as a way to define the security requirement of the DG-Trac project. Therefore, we are not planning to define the security assurance requirements. We will focus on the generation of the security functional requirement part. The proposed methodology that will rely on ADTree will also focus on the security functional requirement.

2.3 Background on ADTree

ADTrees [6] aim at modelling how attackers can attack and exploit weaknesses in a given class of systems. It shows the goals of the attacker in addition to the types of the attack and also implicitly shows the security assumptions that were exploited. Indeed, most attacks are due to unrealistic and unsecure assumptions made by the developers. For instance, a man-in-middle attack exploits the false assumption that the communication channel is secure and nobody can eavesdrop it. This assumption is sometimes false when, for instance, the communication goes through the Internet.
In addition to attack aspects, the attack-defence tree helps modelling the countermeasures to prevent the identified attacks. Figure 2 illustrates how ADTrees are modelled. In it, red circles represent threat goals while countermeasures are by represented by green rectangles. Conjunctive sub-goals refer to sub-goals required simultaneously to reach the parent goal. On the other hand disjunctive goals are independent. For instance as shown in the example bellow, fulfilling only the disjunctive sub-goal 1 is enough to enable reaching the main goal, while conjunctive sub-goal 1 and sub-goal 2 are both required for fulfilling the disjunctive sub-goal 2. Conjunctive and disjunctive relations enable to have a very powerful reasoning tool to model how the main goal can be refined to express both independent and dependent sub-goals, which will be later refined too until reaching the lowest level of detail required to understand how attacks are performed.

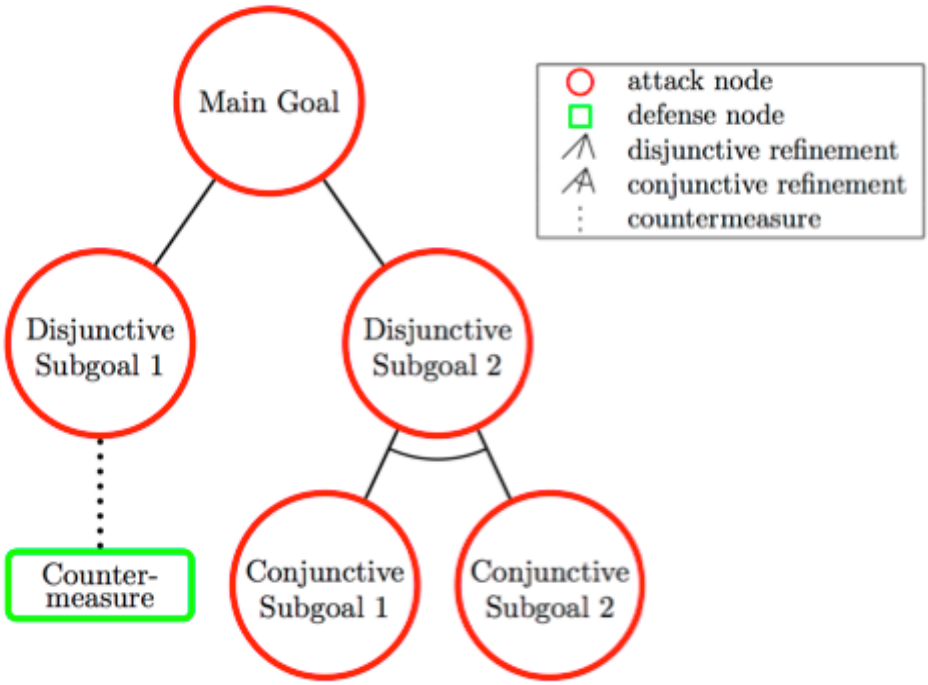

Figure 2 – AD Tree model

2.4 State of the art

This section present the previous work related to PP process improvement and SE techniques based on ISO/IEC 15408. In addition, we will show existing work on security requirements elicitation, which are based on different modeling approaches,

relying mainly an extension of use-cases to include security aspects or attacker point of view.

2.4.1 Improvement of PP process

Williams et al. [7] proposed P3I, an approach for improving the process of creation of protection profiles. They propose to apply System Security Engineering Capability Model (SSE-CMM, ISO/IEC 21827 [8]) to guide the creation of the protection profile.
More concretely, SSE-CMM identifies 11 security engineering process areas. They are described in detail in the methodology. Each process area has some goal (1 to 3 goals) that can be fulfilled by performing the base practices of that process area. The process areas are the following:

1. Administer Security Controls
2. Assess Impact
3. Assess Security Risk
4. Assess Threat
5. Assess Vulnerability
6. Build Assurance Argument
7. Coordinate Security
8. Monitor Security Posture
9. Provide Security Input
10. Specify Security Needs
11. Verify and Validate Security

Williams et al. propose to rely on these processes to build the PP. The idea is to follow one or more process to create each section of the PP document. For instance, to establish the description of the IT environment (How to define the threats and assumptions), they propose the follow the process 2,3,4 and 5. It also will help take into account the impact of threats. They claim that by following these processes, the PP developer will avoid creating poor and costly PP because they will take into account the results of the risk analysis.
For each part of the PP document, they propose to use some of the process areas. To identify requirements, they rely on process 1,7,9 and 10. Interestingly, they propose also an approach for validating the resulting PP. This approach also relies on the SSE-CMM and is based on applying process 6 and 11.
Finally, they motivate their approach by relying on the fact that it is based on CMM, which was first used in the software development area (Software CMM) and had shown to be very effective in improving the quality of software in terms of defect reduction, cost and speed of development.
Their approach is quite interesting because it relies on a well-described and documented methodology. However, they do not provide a detailed, step-by-step process to describe how the SSE-CMM can be reused to guide the PP creation. Their approach could be considered as a set of guidelines and lacks formalization. In addition, they do not provide a concrete example to show how their approach would be applied concretely in practice. An example would be really helpful and would provide confidence in the proposed approach. It will show that it is feasible and can be feasible in practice.

2.4.2 SE techniques relying on ISO/IEC 15408

Substantial research work has been conducted in the area of SE based on the common criteria standard. Most of this work is done by Morimto el a. [9-14]. In [9], he proposed a complete engineering process that relies on CC to guide the design, development, operation and maintenance of secure information systems.
He also used CC to propose a new formal verification technique (based on the Z notation) [13].
All these approaches extend the usage of common criteria and rely on it to perform other engineering tasks.
In this work, we try to use other existing modeling tools to improve the creation of PP, namely the ADTree. For a more general purpose, to define the security requirement, other approaches rely on extending the usage of use-cases. They will be presented next.

2.4.3 Approaches based on extending use-cases

There are several approaches [15-20] that rely on extending use cases to model security. Among them, we can cite the work done by McDermott et al. [17] who proposed "abuse case", which aim at describing the scenarios involving a user (not necessary a hacker) that lead to harming the system security. They reuse the paradigms included in the use case models, which enables them to reusing existing tools to create abuse case diagrams. To illustrate how abuse cases can me created, we present a concrete example (shown in Figure 3 and taken from their paper).

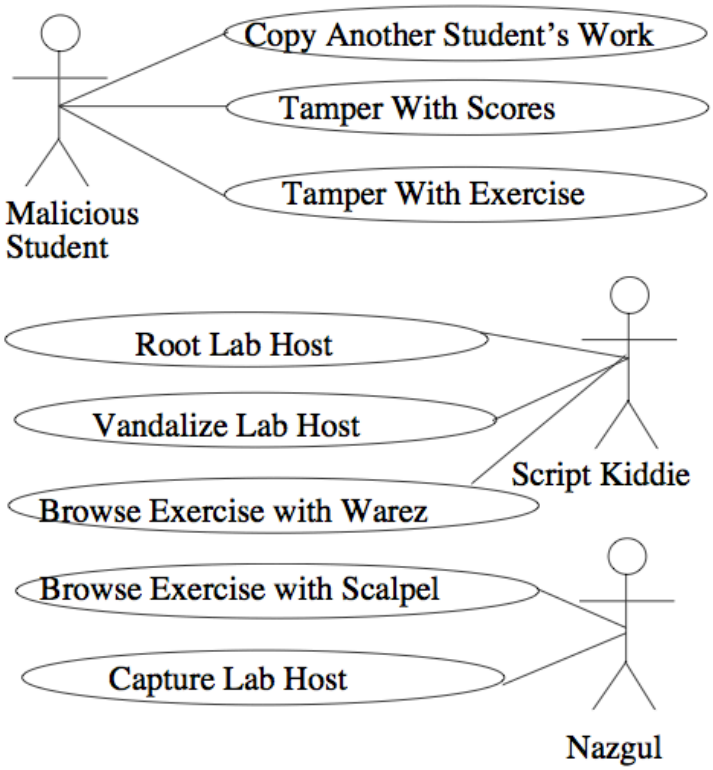

Figure 3 - Example of abuse case

Figure 3 shows an example of abuse case involving three actors (malicious student, script kiddie and "Nazgul"). The context is related to an Internet based information

security lab. The students may abuse the security and try to harm the system as illustrated by the abuse case by; modifying the scores of the lab or copying the work of another student or even changing and modifying the exercise. More advanced attacks could be performed by the script kiddie (as shown in the figure).
With abuse case, it is possible to model several different potential actors of the system. This approach benefits from the simplicity of the use case model to show in a way that is quite easy to understand and communicate the potential security issues that the system could face.

There is another interesting approach promoted by Firesmith [18], who proposes "security use cases". Instead of modeling the attacks, he proposes a new approach oriented toward the defenses. It aims at answering the question related to what security requirements the application needs to successfully mitigate the potential threats. These threats can be modeled using misuse case for instance. This approach is therefore complementary to abuse case or misuse case techniques. The misuse case models can drive the generation of the security use case. It is also complementary to normal use cases, which aim at defining the application main functions and the different usage scenario of the system to be developed. The security use cases will focus only on the security requirements part. It will use the misuse case scenario as basis to help define the security requirements.
In his paper, Firesmith [18] presents several examples of security use case related to Access Control, Integrity and Privacy. He presented several scenarios of misuse and showed for each of them how the security requirements can be defined to mitigate and protection against the threat.
Instead of using the common use case-modeling artifact, he chose to illustrate the security use case using tables. As show in Figure 4, the example of security use case is related to the privacy issue.

Use Case: Privacy			
Use Case Path: System Message Privacy			
Security Threat: The misuser accesses a private message from the system to the user.			
Preconditions: The misuser has the means to intercept a message from the system to the user.			
User Interactions	**Misuser Interactions**	**System Requirements**	
		System Interactions	**System Actions**
			The system shall make the private message unreadable while in transit.
		The system shall send a private message to the user.	
	The misuser intercepts the system's private message.		
Postconditions: The system shall have sent the private message in a form that the misuser cannot read.			

Figure 4 - An example of Security use case

This example tackles the issue that occurs when attackers are able to access private information sent by the system to other users by intercepting the exchanged message. As presented in the figure, the security requirements that are needed to mitigate this threat rely on protecting the exchanged information during its transit. The requirement does not state how this is done. This part is left to the system architect who will choose what appropriate means are needed to fulfill the defined system actions. In this case, this action could be performed using encryption. The information that is sent to the user should be encrypted to avoid this kind of man-in-the middle attack.

All these approaches are quite interesting. They are simple to use and easy to understand and communicate to stakeholders. However, compared to ADTree, we think that the latter provides more expressiveness and is much better modeling tool.
This claim is backed by Opdahl et al. [21] who compared between attack trees and misuse case. In their paper, they performed a pair of controlled experiments, in which they gave a security problem to two groups (students). They were asked to elicit the threats using the two techniques. Their results show clearly that attack trees enable to identify a significantly higher number of attacks. This means that Attack trees have better expressiveness that enables users to identify more threats than when using misuse cases.
In addition, in terms of quality, the results showed that attack tree enable to find the main threats, which represent real security issues for the systems. This means that attack trees provide better results than the misuse cases.

3 The contribution: An ADTree–driven PP creation

The proposed approach relies on defining an attack-defense tree, which will later be used to generate a first draft of the protection profile.
The main steps to be followed are the following:

- **Create an ADTREE:** The user has to define the attack-defense tree. It is important that the attack defense tree do not include low-level details. It should rather be high-level and presents high-level attack goals as well as high-level defenses (Security Objectives).

- **Fill out missing information:** The ADTree is not complete and other information is required to be able to generate the draft. The user has to provide extra information regarding; the Security Functional Requirements attached to the selected defenses also referred to as security objectives, a brief description of attacks (threats).

- **Generate the PP document**: The tool will create the protection profile document based on the tree and the information the user added. This step is fully automated and does not need the user involvement.

- **Check the document and complete it**: This is the last step. The user needs to finalize the documents by adding mandatory sections that were not created during the automated process.

The whole process is illustrated in Figure 5, which highlights all the four steps.

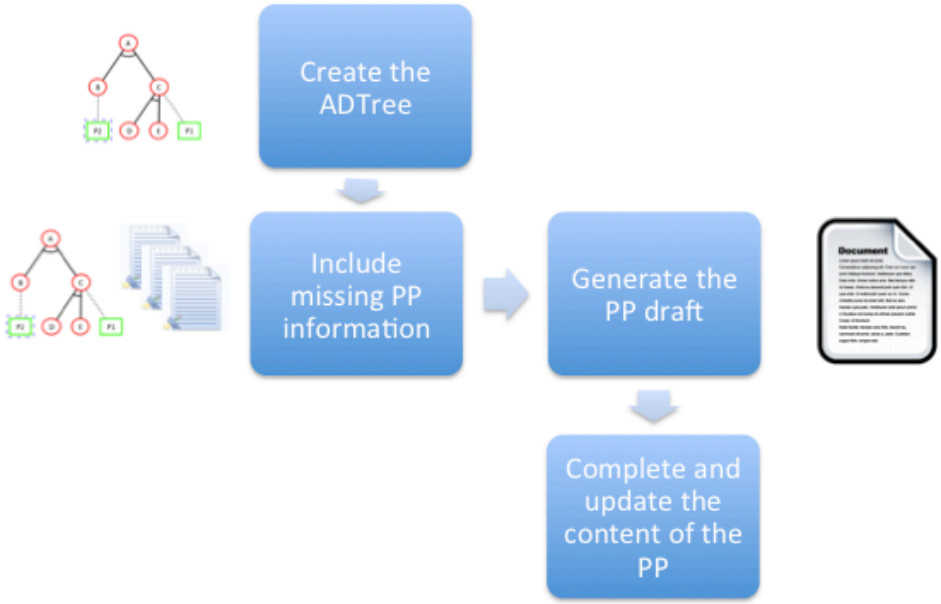

Figure 5 - Overview of the process

As clearly shown in this figure, the first two steps are performed by the user who is in charge of creating the ADTree and adding the additional required information to enable later the automated generation of the PP document. This generated document should not be considered as the final version. In fact, the user needs to review the document and complete it (if needed) and update its content before finalizing it.
This process does not aim at replacing the work of the user. In fact, the user is still in the center of the creation of the protection profile. The process aims at guiding the user by providing an interesting modeling tool that is easy to use, yet powerful and will simplify the communication with the other users/clients or vendors, who are interested in the protection profile.
In addition, the ADTree provides an interesting graphical tool to reason about the threats, the defenses, and more importantly what protections are required to mitigate these threats. The conjunctive and disjunctive relations between threats and defenses is a powerful artifact that can be used to add/compute several attributes related to probability of success, the cost of the attacks and even the power consumption for both the attacks and defenses scenarios. All these factors are important in order to classify the threats in a systematic way. Due to cost constraints, an organization could choose to implement only a subset of the protection mechanisms and will therefore need to know which protection are more important than the others because they help mitigate the most important threats. Importance involves the likelihood of the threats and the impact the threats.
There are still however several issues and problems to be tackled to provide a better process. A nice tool is not sufficient to guarantee a systematic and effective approach that will produce high-quality protection profiles.
The main issue is related to the completeness of the ADTree. It is difficult to guarantee that all the threats were taken into account and that the ADTree has considered all potential threats that could occur. We believe this is a hard and difficult and is still an open question in the area of security requirements elicitation. The common approach relies on considering all the existing and known threats, while unknown threats and new attack techniques are by definition difficult to predict. This explains the reason why even the most protected systems are still vulnerable to attacks and involve security breaches no matter the amount of money spent to secure them. Highly secure systems were found vulnerable because they do not protect against all possible and potential attacks. We will try to tackle this issue and provide an answer (although partial).
The second issue is related to the degree of detail to be considered. On one hand, protection profiles do not go into the details and remain always at high level. On the other hand, ADTree are considering very specific aspects and each added sub-level provide more detailed view on the threat of the attack. ADTree provides a low level view, which is too detailed and cannot be added directly in the protection profile document. This problem will also be considered and tackled by the approach.
This remainder of this section is organized as follows. In the first subsection, we provide a detailed description of the four steps of the process and explain how the raised issues are tackled. In the second subsection, we present the tool implementing the approach. In the third subsection, we present how the approach is applied to the context of DG-Trac to assess the feasibility of the approach. In the last subsection, we will discuss the threats to validity of our approach and the main limitations of the technique that we are developing.

3.1 Step-by-step detailed view

As shown in Figure 5, the process involves four steps. They will be presented, explained in detail in this subsection.

3.1.1 Creating the ADTree

The creation of an ADTree is quite simple, straightforward and intuitive. The user has to define the main threat goal, which will then be refined into sub-goals. All potential threat goals should be considered and more importantly all possible ways to achieve them should be taken into account.

As pointed out previously, the main problem in defining the ADTree is how to guarantee the completeness of the approach. To tackle this issue, the solution that we propose is to follow the best practices to identify the threats related to the area. We propose that the user relies on two specific sources to select the threats; the SHIELDS SVRS [22] and the ECMA PP E-COFC public business class [23].

Firstly, let us consider the SHLIEDS SVRS. SHIELDS project is a European project funded through the FP7 (2007-2013). Its main goal is to provide developers with a repository that contains a knowledge database about; security threats, security vulnerabilities, security activities (what concrete activities are needed to secure the system), causes (what causes the vulnerability). SHIELDS provides this knowledge database in a security vulnerability repository service (SVRS), which is accessible online[1]. We are particularly interested in both the threats and the security activities. Vulnerabilities are the set of potential flaws that an attacker will try to exploit in a particular system (for instance a web application could be vulnerable to Cross-site scripting attacks). Threats are more interesting because they represent the attack goal, which an attacker tries to reach by exploiting one or more vulnerabilities.

Figure 6 illustrates some examples of threats that the user can find in the SVRS website (it is taken from the website). Threats include physical attacks like for instance physically accessing into the computer as well as remote (traditional) attack like breaking encryption etc.

The SVRS contains 287 threats, 130 security activities, 30 security goals and 399 causes of vulnerabilities. This repository is quite interesting to use as a basis by the user to create the ADTree. The user can follow the following scenario:

- Describe in detail the assets of the systems.
- For each asset consider how users interact with it.
- Depending of the underlying technology/systems used to interact with the assets, check which threats apply to the system.
- Consider the security objectives (Confidentiality, Integrity and Availability always referred to as CIA) when thinking about possible attacks to be taken into account. Then based on the repository select the security protections (activities) to be in place to protect the system assets.

Figure 7 shows some example of security activities included in the SVRS.

[1] https://svrs.shields-project.eu/SVRS/

Categories	Causes	Security Activities	Security Goals	Threats	Vulnerabilities

Name	Description	Author	Last updated
Break into computer		PH Meland	2010-6-16 21:5:30
Break into machine and read unencrypted key out of memory		PH Meland	2010-6-16 13:25:44
Break into the download site		PH Meland	2010-6-16 22:5:51
Break into the machine and read encrypted private key off disk		PH Meland	2010-6-16 13:25:41
Break network encryption	Get access to a network connection by breaking the...	PH Meland	2010-6-16 11:55:21
Break password file		PH Meland	2010-6-16 21:15:6
Break RSA encryption		PH Meland	2010-6-16 12:49:42
Break the public key encryption	Break the public key encryption	PH Meland	2010-6-16 12:49:41
Break the symmetric key encryption		PH Meland	2010-6-16 12:55:40

Figure 6 - Examples of SVRS listed threats

It is important to consider these guidelines as a reference and that the threats (attacks) and the security activities (the defenses) must be adapted to the context of the system. Each system has its own specific characteristics that need to be taken into account. There are always attacks that are specific the system and that will not be in the SRVS list. The user has to be aware of this aspect to avoid ending up with a generic ADTree that will not tackle the specific problems of the system that will surely be exploited by attackers.

Name	Description	Author	Last updated
Encrypt data during storage	Information can be encrypted during storage to mak...	PH Meland	2010-6-10 16:55:10
Encrypt data in transit	By encrypting data in transit it gets harder for a...	PH Meland	2010-6-10 16:55:9
Enforce proper access control	Control who gets access to resources, and make sur...	PH Meland	2010-6-8 14:23:37
Enforce user authentication	Make sure all users are authenticated and that the...	PH Meland	2010-6-8 14:23:37

Figure 7 - Examples of SVRS listed security activities

The second main reference to be used when creating the ADTree is the ECMA PP E-COFC public business class [23]. ECMA is the European association for standardizing information and communication systems. It is a private standards organization. In 1993, this organization has published a standard ECMA-205 “Commercially Oriented Functionality Class for Security Evaluation (COFC)” [24]. In this standard they focused on commercial requirements and provided a set of functional criteria to evaluate the security. Later, in 1999, they extended this standard

to include interconnected systems. This led to a new standard, ECMA-271 "Extended-Commercially Oriented Functionality Class for Security Evaluation (E-COFC)" [25]. To match the Common Criteria methodology, a protection profile was then, created, which completes the functional criteria, with the assurance criteria. This protection profile is quite interesting because it considers commercial requirements of the organization, which will be suitable for commercial systems, which are not security products (like firewalls, IDS, antivirus). This protection profile matches the need of usual companies, which are developing commercial products, which are not security-critical systems.

The E-COFC protection profile follows the common criteria methodology and provides a description of the TOE environment that includes a set of threats and assumptions. They also define a list of security requirements and security objectives, which fit the context of commercial systems.
Figure 8 shows some examples of the threats identified in the E-COFC PP (it is taken from the E-COFC PP document).

	Threat name	Threat description
19	T.Outsider	An individual who is not an authorized user of the system may gain access to the TOE.
20	T.Physical	A component of the system may be accidentally or intentionally damaged.
21	T.Privacy_Violated	Unauthorized access to privacy data of system users may occur without detection.
22	T.Refusal_Undetected	Unauthorized refusal of valid commitment data or certificates may not be detected.
23	T.Replace_TOE	The TOE may be replaced by an untrusted system.
24	T.Replay	Someone may obtain unauthorized access by replaying authentication or commitment data.
25	T.Secret_Disclose	Authentication information may be disclosed allowing someone to logon and assume the identity of an authorized user.
26	T.Service_Denied	Application and network services may not be available for use.
27	T.TOE_Fail	TOE or system failure may cause the TOE to enter an insecure state or data to be disclosed or changed.
28	T.Traffic	A system may experience degraded performance due to increased communications traffic.
29	T.Unique_Copied	Unlawful copies may be made of unique originals.

Figure 8 - Examples of E-COFC PP threats

We consider this PP as an excellent reference for the users who will create the ADTree. It should be used as a starting point because it shows what are the threats that should be considered when dealing with commercial systems. They consider aspects related to the customers, quality of service that is important for any commercial service.

However, it should not be taken for granted and completely reused to create the protection profile. The user should be careful and should always adapt the ADTree to the context of the system to be secured. Each system has its own characteristics that should be taken into account when identifying the threats it faces.

Once the threats identified, the user has to build the ADTree and defines the relations between the attack goals and sub-goals by using the disjunctive and conjunctive refinements. Each attack should be refined into sub-goals. Then, to tackle the threats the user should choose a set of protections that need to be used to protect and help mitigate the attacks.
The second main issue to be handled is about how to bridge the gap between the ADTree low level details and the protection profiles, which present high-level view of the security requirements. To solve this issue, we recommend that the user considers only high-level aspects and does not produce low-level detailed ADTrees. Based on the use of E-COFC PP, the resulting ADTree will be high-level and will match perfectly the required level of abstraction that is needed in the protection profile.

3.1.2 Completing ADTree with more information

This step requires that the users complete the ADTree with the missing details needed to enable the automated generation of the PP document:

1. Description of the TOE: A description of the system to be secured, its main functional requirements and what are the characteristics shared by the PP targeted systems.
2. Security Functional Requirements: these requirements are based on the common criteria methodology document (Part 2: Security functional requirements [26]). These security functional requirements are related to the TOE but also to the TOE environment.
3. Rationale between Security Functional Requirements and Security Objectives: This rationale is a mapping that explains how SFR are able to cover and fulfill all the security objectives identified previously.

Only part 2 and part 3 are needed to generate the PP document (the security functional requirements and the mapping between the security requirements and the security objectives). The description of the TOE can be added later directly to the PP document.
The mapping between the security objective and the security requirements can be obtained by choosing among the requirements available in the Common Criteria document (Part 2) which requirement matches the security objective that is chosen in the ADTree. In this document, the security requirements are organized according to their class. 11 classes are presented by the common criteria part 2 document. This document provides also documentation about the objectives of each security functional requirement. The annex provides quite interesting information about the way the security requirements can be used when defining a new security target or a protection profile.
It is also important to note that each security objective can be mapped to several security objectives and the latter can be associated to several security objectives. In

fact, this can be explained by the fact that an objective can sometimes be reached by combining several requirements to help fulfill the chosen security objective.
In addition, it is important to define security requirements and match them to both the environment and the TOE. This depends on the type of the security objective. This will have an impact on the generated document because there are two different sections for security objectives/requirements targeting the TOE and the TOE environment. Therefore, the user should define which security objective is related to the TOE environment and which security objective is related to the TOE.

3.1.3 Generate the PP draft document

This section explains how the generation of the PP document is performed by our approach. This step is completely automated and represents the core of the tool that is developed.

We take as input the ADTree created by the user and the additional information provided by the user concerning the mapping between security requirements and the objectives. The most important step in this automated process is related to the strategy to adopt on how to use the ADTree.

As presented in Section 2.3, the ADTree is composed of conjunctive and disjunctive refinements. Several defenses (security objectives) can be associated to any node or sub-node. To select the defenses to extract from the tree, we need to choose which one is mandatory to select and which ones are optional because there are other defenses in place that could be used instead of the other defenses. To illustrate this issue, let us take as example the ADTree shown in Figure 9.

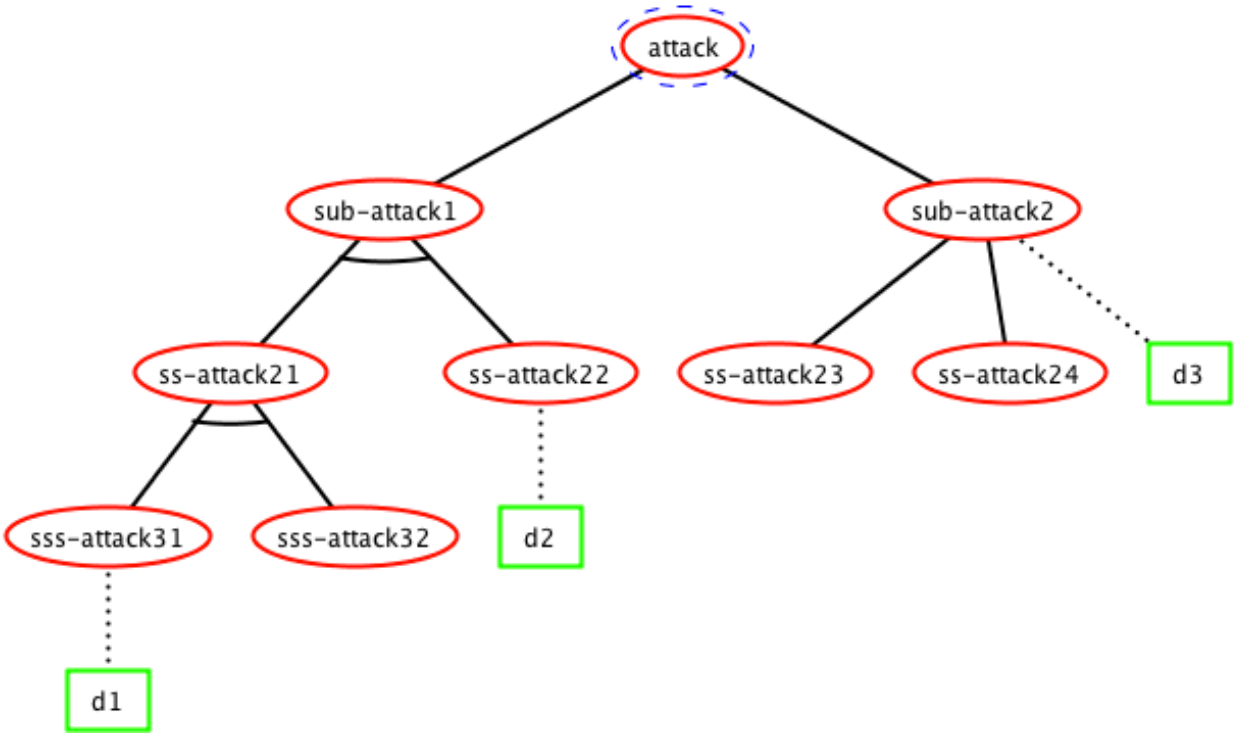

Figure 9 - Example of ADTree with disjunctive and conjunctive refinements

In this simple example, there are only tree defenses. As shown, these three defenses enable to protect against the main attack called 'attack'. Since the relation between the sub-attack1 and sub-attack2 is disjunctive, this means that we need to choose defenses protecting against both sub-trees attacks. However, when we consider the sub-tree under sub-attack1, we notice that there is a conjunctive refinement. Therefore, in this case, we have two options. We can either select the defense against "ss-attack22" (the "d2" defense) or select the protection against ss-attack21 that will be "d1" (countermeasure of the sss-attack31). We can also choose both to guarantee more security and make sure to protection against sub-attack will work by using redundant security mechanisms. However, security comes with additional cost that may not be accepted due to limited budget.

To sum up the issue, we define three strategies to cope with the problem of conjunctive sub-trees:

- Strategy 1: Based on considering only one sub-tree defense and ignore other sub-trees. The issue with this strategy is the following: What if the security protection fails for that sub-tree? The security protection may fail which will make the attacker succeeds in reaching his goal.
- Strategy 2: Consider all sub-trees defense. This strategy will tackle the issue raised in the previous strategy. However, this solution will lead to higher costs, which could not be accepted by the organization that will implement the solution.
- Strategy 3: Consider only subset of the trees and select the defenses according to their cost. The overall cost of the protections should not exceed an amount defined by the organization. If this amount is not reached yet, we can select more that one protection for conjunctive refinements. In this case also, the user needs to define for each attack a value to make a ranking of attacks. This is important to enable selecting which sub-attacks to protect against first.

As explained in Figure 10, which presents the first strategy algorithm, the idea is the take into account only the first defense of conjunctive refinement sub-tree. When there are conjunctive refinements, the algorithm calls a function that will go through the sub-tree to locate the defense(s) of one of the sub-tree (and ignore the other defenses). They could be more than one defense when the sub-node is itself decomposed into disjunctive sub-nodes (in this case there are more than one defense needed).

```
23      public void strategy1(ADTreeNode node, List<Defense> defenseList) {
24
25              while (node.hasSubNodes()) {
26
27                  if (node.getRefinementType().equals("CONJUNCTIVE")) {
28
29                      List<Defense> defenses = retrieveFirstDefensesOfSubnode(node);
30                      defenseList.addAll(defenses);
31
32                  } else {
33
34                      for (ADTreeNode subNode : node.subNodes()) {
35
36                          if (subNode.hasDefense())
37                              defenseList.add(subNode.getDefense());
38
39                          strategy1(subNode, defenseList);
40                      }
41                  }
42              }
43      }
44
```

Figure 10 - Strategy 1 algorithm

Figure 11 presents the algorithm for the second strategy. As shown, it is clearly the easiest strategy to implement since it takes into account all sub-trees and collects all defined defenses (in the ADTree).

```
14      public void strategy2(ADTreeNode node,List<Protection> protectionList) {
15
16          while(node.hasSubNodes())  {
17
18              for(ADTreeNode subNode: node.subNodes()) {
19
20                  if(subNode.hasDefense())
21                      protectionList.add(subNode.getProtection());
22
23                  strategy2(subNode,protectionList);
24              }
25          }
26      }
27
```

Figure 11 - Strategy 2 algorithm

As shown in the algorithm, all defenses are retrieved without checking any other information regarding redundancy and the type of refinement (conjunctive or disjunctive).
Finally, Figure 12 shows the algorithms of the third strategy. It takes into account the attributes given to classify the attacks and to give cost to the defense mechanisms.

```
46      public void strategy3(ADTreeNode node, List<Defense> defenseList) {
47
48          while (node.hasSubNodes()) {
49
50              if (node.getRefinementType().equals("CONJUNCTIVE")) {
51
52                  List<Defense> defenses = retrieveFirstDefensesOfSubnode(node);
53                  defenses = orderDefensesByCostAndByAttackRanking(defenses);
54                  defenses = selectSubsetOfDefensesBasedOnAttributes(defenses);
55
56                  defenseList.addAll(defenses);
57
58              } else {
59
60
61                  for (ADTreeNode subNode : node.subNodes()) {
62
63                      if (subNode.hasDefense())
64                          defenseList.add(subNode.getDefense());
65
66                      strategy2(subNode, defenseList);
67                  }
68              }
69          }
70      }
```

Figure 12 - Strategy 3 algorithm

It is important to note that the choice of the strategy to use is left to the user. In fact, the user may want to choose strategy 1 to have better security. In this case, the cost is not an issue. He may also want to use the second strategy if he wants to have the less expensive solution. Finally, the third solution is also interesting. However, it requires the user to provide the cost of each defense mechanism, which may be difficult in

some cases. He also needs to provide a classification of the attacks. This step is also difficult to perform. The best method to enable classifying the threats is to rely on statistics about the current threats. These statistics could be retrieved from online repository like the CVE (Common-Vulnerabilities and Exposures [27])

3.1.4 Update and finalize the draft

This is the last step of this process that the user needs to fulfill manually. It is up to the user to check and update the final version of the PP draft.
In fact, there are some sections that cannot be automatically retrieved from the ADTree or from the additional information that the user provides.
The user needs to fill the assumptions section. In it he has to provide the list of assumptions that are taken into accounts. In many existing protection profiles, we notice that for IT Systems the following assumptions are usually used:

A.PHYSICAL - It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information. The physical access to DG-Trac assets is protected against physical attacks.

A.TRUSTED_ADMIN - It is assumed that the DG-Trac administrators are to be trusted, appropriately trained and follow administrative guidance and do not seek to harm the system or perform attack aiming at disclosure of confidential information.

A.FIREWALL - The firewall is assumed to be configured as the only network connection between the private network and the hostile network.

A.PEER - Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints.

A.USER - Users of the TOE are assumed to possess the necessary privileges to access the information managed by the TOE.

These assumptions are always used in many protection profiles. They are important to enable the PP to focus on the most important part, which is securing the system itself. In most cases, trusting the administrators of the system and guaranteeing that the system will be hosted in a secure and safe place is mandatory because securing against these two threats is difficult and quite costly, and more importantly is unnecessary (for many organizations). In addition, it is important that the users have the right to access the TOE information. In addition, the firewall assumption is important to avoid having the TOE to tackle the issues of network level attacks, which have impact on the whole organization. This goes beyond the scope of securing the TOE and is related to securing the organization network itself. Finally, the peer assumption is important because the objective is to secure the TOE and not the secure all the others systems that the TOE communicates with.

In addition to the assumptions part, the user needs to fill the organizational security policies part. In this section, as noted in many protection profiles, this part is in most cases related to access control that is enforced.
Finally, there are some other mandatory sections, which are not generated by our approach and need to be filled by the user: The PP introduction (identification and overview); the TOE description and PP application notes.
The Security Assurance Requirement part should also be filled by the user (depending on the chosen evaluation assurance level).

3.2 The tool supporting ADTree-Driven PP creation

A tool was implemented as a plugin for the ADTool [28]. This tool was previously developed by one of the SnT research groups called Satoss (led by Prof. Sjouke Mauw). This tool allows users to model and display attack–defense scenarios. It was developed within the ATREES project. In addition, ADTool allows performing quantitative analyses on ADTree by defining attributes for nodes to reason about:

- ✓ The costs of attacks and defenses.
- ✓ The attacker skill levels
- ✓ The probability of success of attack/defense scenario.

The tool developed for implementing our approach was included as plugin in the ADTool. It provides a GUI that allows the user to perform the four steps of the approach (as shown in the previous section). The first step involving the definition the ADTree is already fulfilled by ADTool. Figure 13 shows a simple example of ADTree.

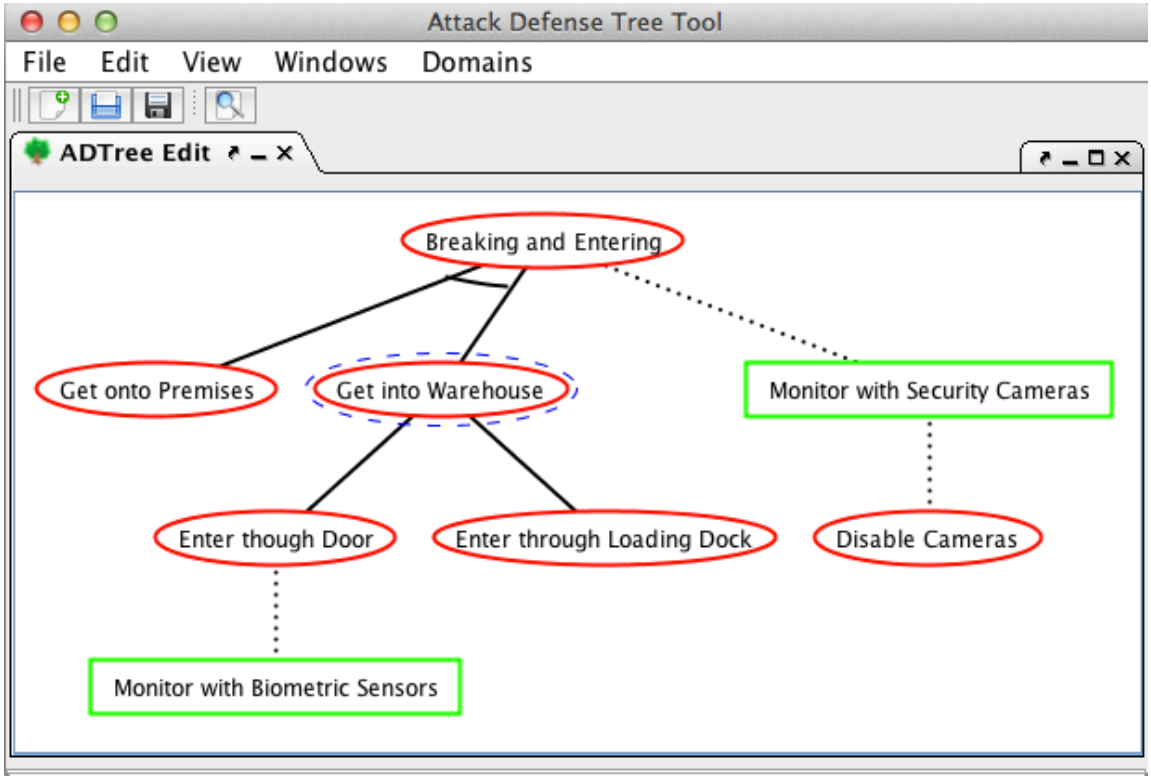

Figure 13 - ADTool main window

The second step is provided by the plugin. The user is asked to fill in the missing information; including the descriptions of the threats, the security objectives. He has also to define a mapping between the security objectives and the security requirements as specified by the Common Criteria part 2 [1]. In the GUI, the user is provided with the list of security requirements and has to choose for each security objective the related security requirements. The tool provides three GUIs to enable performing this task. The user can edit and save the changes. Note that the descriptions and the mapping provided by the user can be saved in a file (with ".pp" extension) and loaded later to perform the PP generation process. This option requires

the user to save this data in a file before closing the ADTool application. The user is asked whether he wants to save the PP file before closing the application.

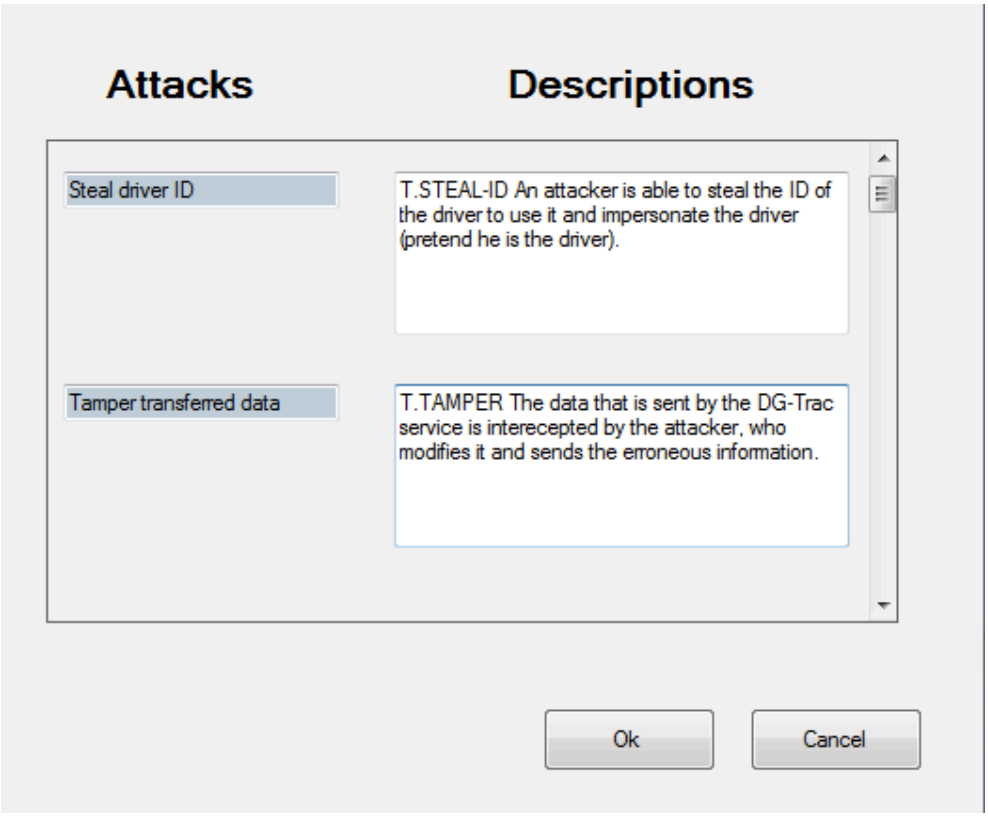

Figure 14 - GUI for adding attack description

Figure 14 shows the description of the attacks (threats) that could be added inside the tool. It is important to note that the user should include an acronym having the following format "T.NAME_OF_THE_THREAT" or "TE.NAME_OF_THE_THREAT" when providing the description (as shown in the figure T.STEAL-ID and T.TAMPER). This is important in order to follow the naming convention as described with the PP guidelines. These descriptions will be later used to generate the PP document. Note that the threats related to the TOE Environment should have acronyms starting with TE. This is important because it will be used later by the tool to separate the threats related to the TOE from those related to its environment. Each one is put in a different section of the PP document.

Figure 15 shows how to define the mapping between the security objectives and the security requirements (based on the Common Criteria part 2 document). The user has to select for each security objective, which security requirement will be associated to it. The user is provided with a list that is organized following the CC document classification of the security requirements. The user cannot choose the requirement class. He has to choose the requirement sub-class for each security objective. The mapping information is then saved in a file and can later be edited by the user.

The main difficulty is in choosing the right security requirement that corresponds to the chosen security objective. This is of course based on the assumption that the Common Criteria document covers all existing security requirements. The user should be able to fulfill any security objective by relying on the security requirements as defined in the CC part 2.

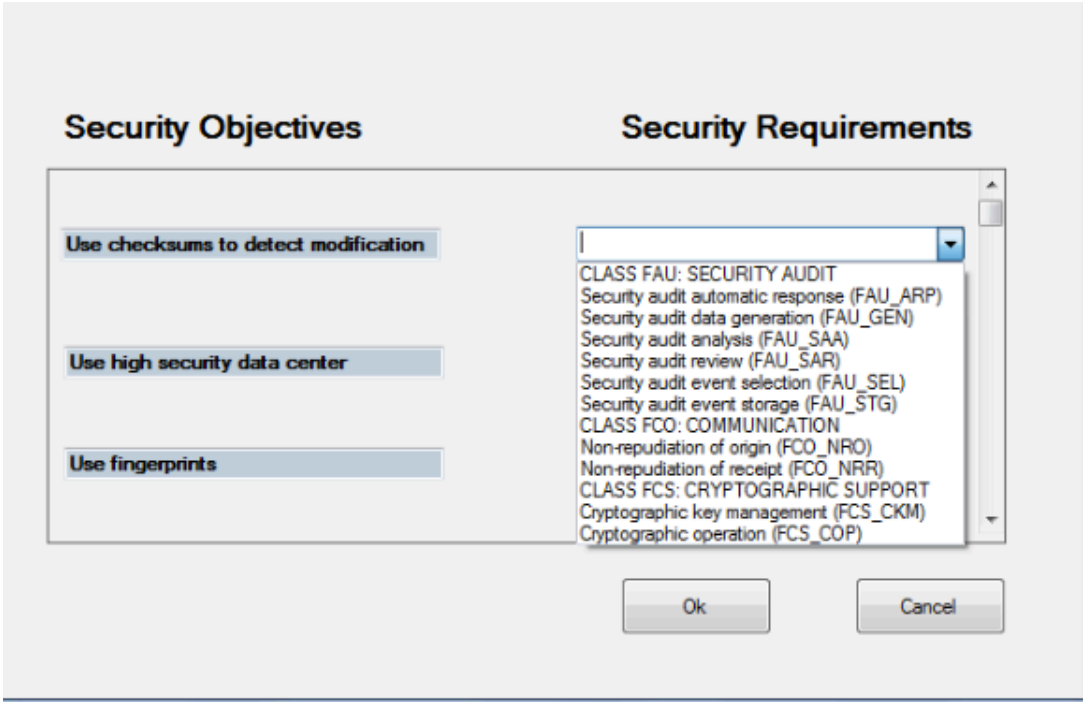

Figure 15 - Mapping Security objectives to CC security requirements

3.3 Applying the approach to DG-Trac case study

In this section, we apply the approach to DG-Trac case study in order to create a new protection profile for dangerous goods tracking and tracing systems. We will follow the strategy explained in Section 3.1. Firstly, we will show the ADTree that was defined for DG-Trac. Then we present the steps followed to complete the obtained ADTree with the missing information needed to perform the automated generation of the PP document. Finally, we present how the draft PP was completed and finalized.

3.3.1 ADTree for DG-Trac

The next figures present the ADTree we defined for DG-Trac. For the sake of readability, the tree is divided into 5 sub-trees.
To build this tree, we followed the guidelines as described in the previous section. Using the ECMA PP E-COFC and the SHIELDS SVRS.

In addition, to make sure that the proposed tree matches the reality; we discussed the proposed ADTree with the DG-Trac project partners. They all confirm that the selected threats correspond to their potential/existing issues and agreed to adopt the proposed tree.

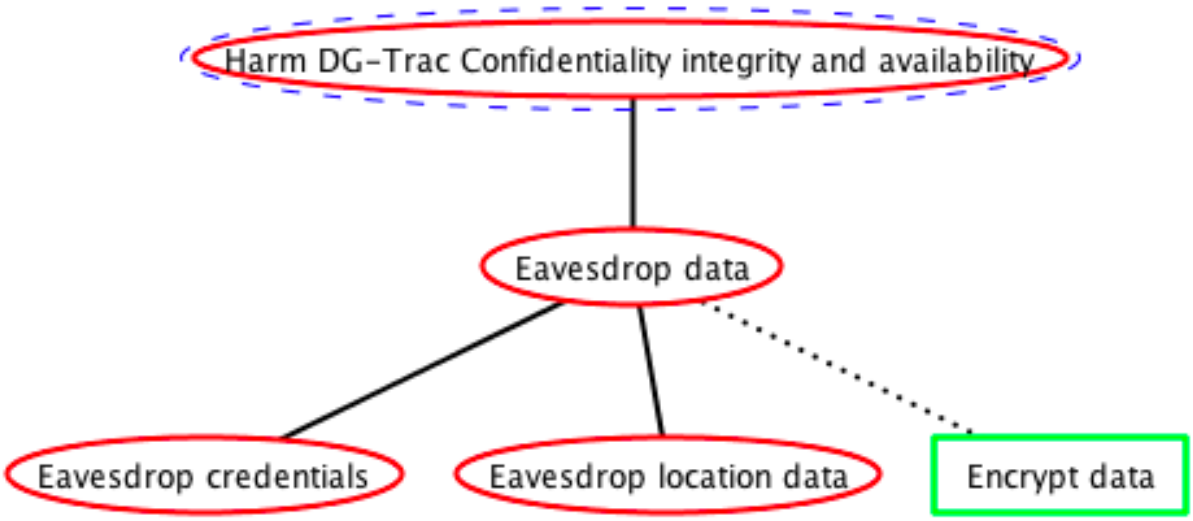

Figure 16 - DG-Treat: Eavesdropping data

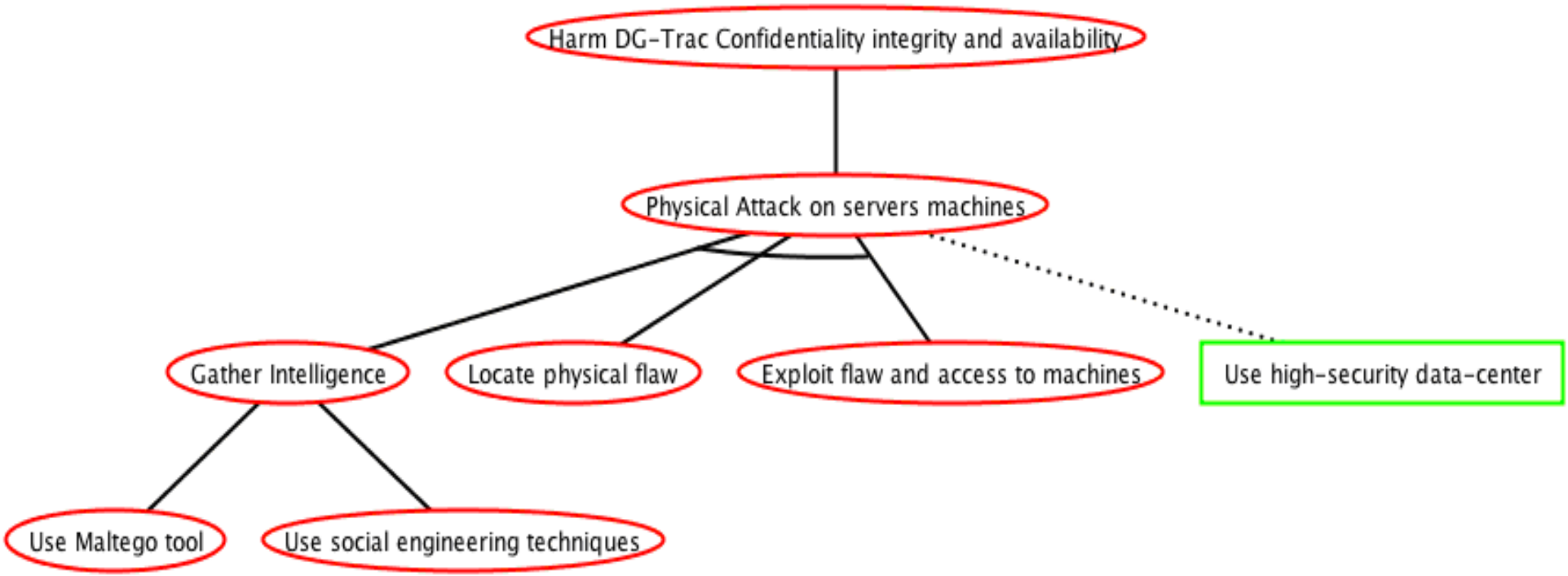

Figure 17 – DG-Trac Threat; Physical Attack on Servers

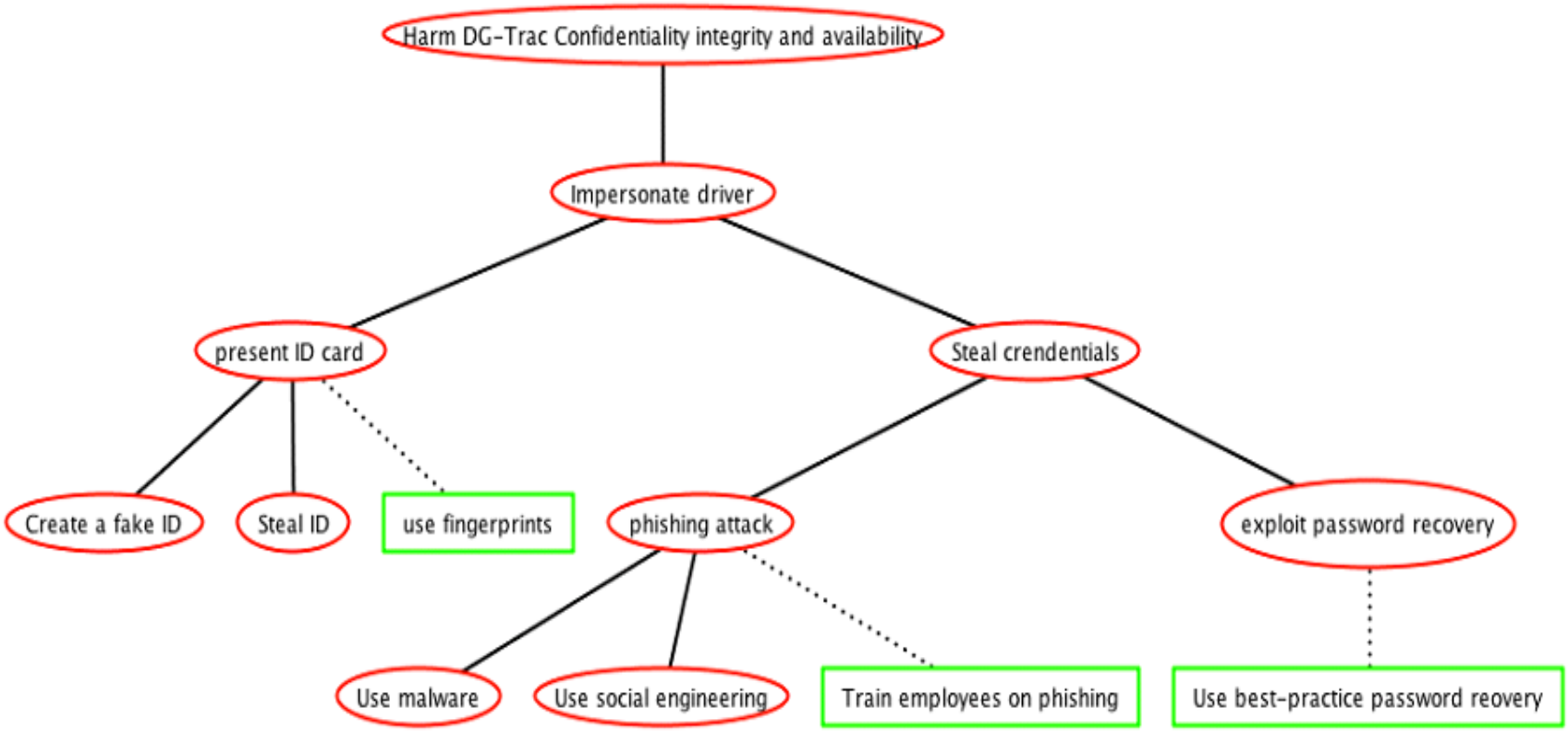

Figure 18 – DG-Trac Threat; Possible Driver Impersonation

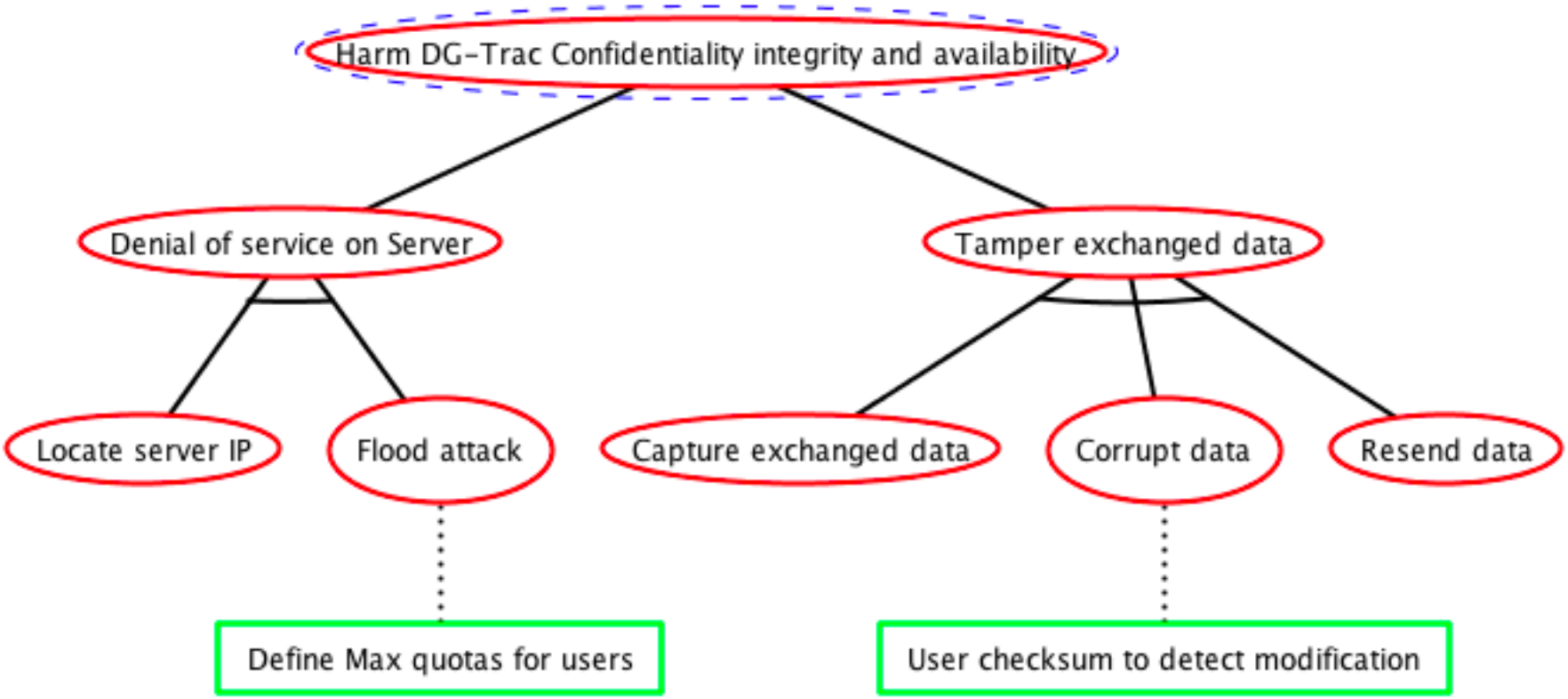

Figure 19 – DG-Trac Threat; DOS Attack and Tampering

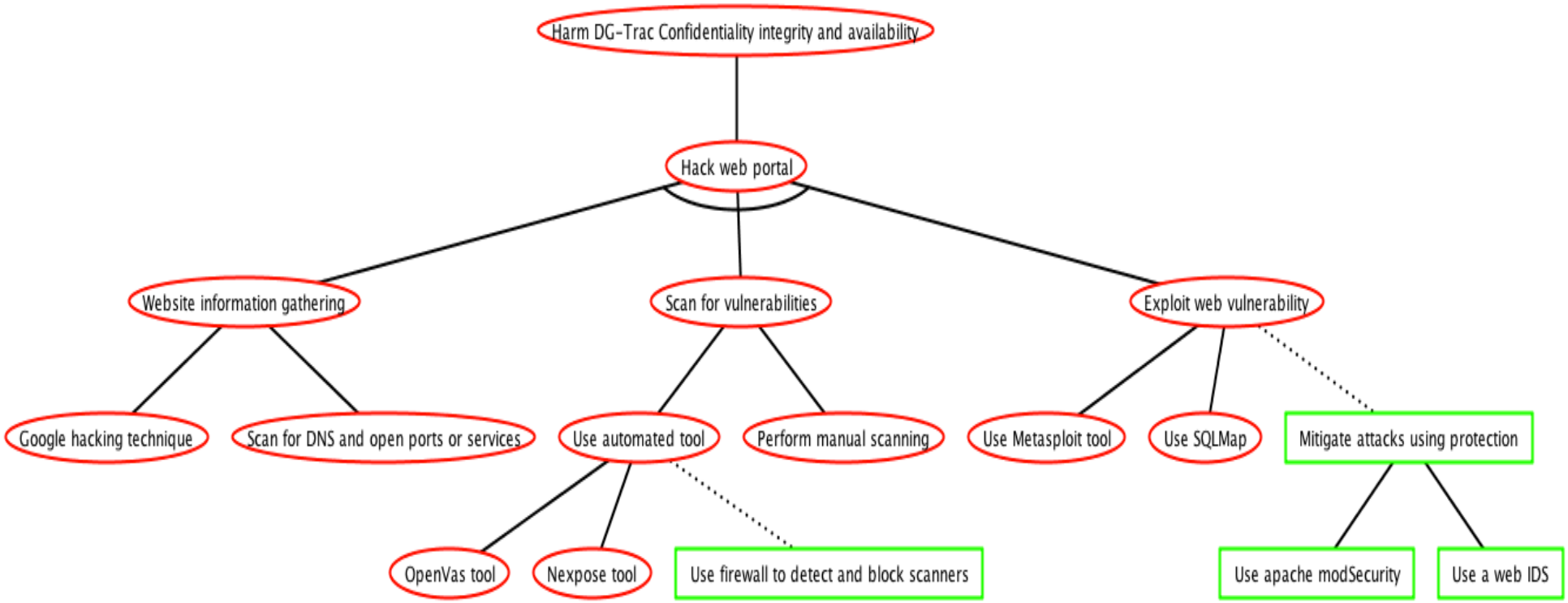

Figure 20 - DG-Trac Threat; Hacking the DG-Trac Web-Portal

As shown in the figures, we identified these six main goals for attackers:

- **Hacking into the web portal**: By exploiting application-level vulnerabilities in the DG-Trac web-portal.
- **Physical attack server machines:** (Figure 17) the attacker tries to hack into the building hosting the DG-Trac solutions to get direct access to machines.
- **Impersonating driver:** (Figure 18) the attacker tries to whether fake a driver ID or to steal his real credentials. The attacker impersonates a genuine driver to get the dangerous goods.
- **Performing a denial of service (DOS**): (Figure 19) the attacker tries to shut the machine by flooding it with huge number of request to disable it. This will cause an interruption of the service during the flooding.
- **Tampering with exchanged data**: (Figure 19) In this case, attackers want to corrupt exchanged data by inputting incorrect information thus making the message completely unreadable.
- **Eavesdropping on data**: The attacker tries to monitor exchanged data and tries to gain access to either credentials (login/password or session key, identification key etc.) or the location data of the transported goods.

In order to mitigate these threats, the following list of counter measures was defined:

- **Encrypt Data:** To prevent that someone intercept and access to private data when send to DG-Trac customers
- **Define Max quotas for user**: This will prevent attackers from abusing the resources and performing a DOS attack.
- **High-security data-centre**: This will protect from physical attacks.
- **Fingerprints:** Useful to avoid faking IDs. Checking the driver fingerprint will make it difficult for attackers to impersonate the driver.
- **Best-practice password recovery:** This will enable avoiding credentials theft by exploiting weak lost-password recovery procedures.
- **Checksum to detect unauthorized modification:** This will guarantee integrity of transferred data.
- **Train employees on phishing**: To prevent employees from disclosing their credentials for instance due to for instance email phishing attacks.
- **Firewall to detect and block scanners:** The scanners are important to automate the detection of flaws. By blocking them using web-firewalls, one can make it difficult for attackers to find weak points and vulnerabilities. However, they can still perform manual scanning, which cannot be stopped.
- **Apache ModSecurity or commercial web IDS**: ModSecurity is a widely used protection tool. It is an open-source tool that enables to detect and block most-common web-attacks. Instead of ModSecurity, any commercial tool can also be used (to have better results).

3.3.2 Mapping the security requirements to the security objectives

Next, we present how the mapping between the objectives and the requirement is defined for the DG-Trac project. Since training employees, and using high security data-center go beyond the system security requirements, they were not included here.

Objective	Security Requirements and justifications
O.ENCRYPT	**FDP_UCT.1** – User data confidentiality (during data exchange) through encryption to guarantee non-disclosure of the information during the transfer.
O.WEB_IDS	**FAU_SAA.1** Potential violation analysis, basic threshold detection on the basis of a fixed rule set is required.
O.FINGEPRINTS	**FIA_UID.2** - The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
O.DEFINE_MAX_QUOTAS	**FRU_RSA.1** Maximum quotas, provides requirements for quota mechanisms that ensure that users and subjects will not monopolize a controlled resource. **FDP_ACC.1 – Subset access control**: the system shall enforce the access control on list of subjects, objects, and operations among subjects and objects of the DG-Trac system. This is needed to enforce max quotas policy depending on roles. **FDP_ACF.1 – Security attribute based access control:** The system shall enforce the access control to DG-Trac service based on the users roles. Roles should be defined for each type of user and access rights (permissions) assigned to these roles according to the required resource and quotas of use.
O. CHECKSUM	**FDP_UIT.1** - Data exchange integrity addresses detection of modifications, deletions, insertions, and replay errors of the user data transmitted. **FDP_SDI.2** – Detection of errors in stored data in the first step to guarantee the detection of integrity problems. **FDP_DAU.2** – Generation and verification of checksums, one-way hash, message digest. All these checks are important to help detect any problem of loss/modification of the user data or also any tampering of data.

3.4 Threats to validity and limitations

There are several limitations and threats to the validity of the proposed ADTree-driven approach for creating protection profiles.

The first threat concerns whether this approach can be applied to others case studies. Applying this approach on only one case study is not enough to prove that this approach could be generalized and applied to other case studies. To tackle this issue, we need to try and apply the approach to several other case studies, having different characteristics, requirements, to guarantee that the approach could fit different contexts. The tool implementing the approach, once publicly available online and shared could be used by other organizations. This will help to study whether the approach could be or not applied to their contexts.

The second threat is related to the scalability. The DG-Trac case study is of a reasonable medium size. The approach needs to be applied to other bigger case studies. The main issue that we foresee, when tackling big case studies is related to the size of the ADTree. A quite big tree will harm readability and make the big picture of the threats/defenses difficult to understand. In this case, there is a need to divide the tree into several sub-trees. In addition, the tool should be adapted to enable merging several ADTrees to generate a unique PP documents. The original tool should be adapted to support displaying several trees in the same window but we think that this s straightforward and will not be difficult to implement. When defining the sub-trees, the user should be careful to avoid redundancy in the threats. The same threat should not be defined in more than one sub-tree to guarantee that the merged threats is consistent and does not contains redundant concepts.

Finally, the last threat to validity is related to the soundness of the approach, from a conceptual point of view. A skeptical user might think that attack-trees concepts (attacks and defenses) are far away from the PP concepts (the security objectives and the security requirements). We think that in practice, we can assume that the user can consider the concepts as similar. The solution is to consider ADTree as TSOTree (Threat-Security-Objective Tree). This will only lead to one limitation. Existing ADTree cannot be reused to create protection profiles. However, this issue is outside the scope of this paper.

The main limitation of this approach concerns the degree of automation, which could be seen by some users as not enough to replace a fully manual creation of the PP document. Indeed, there are several parts of the documents that need to be added manually by the users, including the descriptions of the threats (attacks in the ADTree), the security objectives (defenses in the ADTree) and defining the mapping to the security requirements, among others. We think that a user could still write the full document manually. The approach aims at offering a new process that is supported by a tool. The core contribution is in the process, not in the fully automated tooled approach. Reaching a full automated is an open research issue. We think that it can only be reached by relying on a huge knowledge database. Then, we need to apply applying approaches to analyze natural language requirements documents. There are some quite recent approaches that enable (to a certain level) reaching this full automation. For instance, there is the work done by Xiao et al. [29] in the context of access control policies.

4 Conclusions and Future Work

This work addressed the issue of improving the process of creating protection profile. The aim was to provide a process that is supported by a tool to assist the creation of the protection profile. This process relies on ADTrees to model the threats that a system is facing. These threats are tackled using some defenses. The structure of the tree and its semantics provides a powerful tool to express relations between threats (using conjunctive and disjunctive relations to combine sub-goals). The approach enables the user to create an ADTree that model the threats and the security requirements. Then, the user has to add the missing information needed to generate a first draft the PP document. The user will finalize this first draft and complete it or update its content if necessary.

In this work, we successfully applied this approach to the context of the DG-Trac project. The results show that the approach is feasible and provides confidence that the approach is applicable and useful in practice. The protection profile created by this project will be included in one of the deliverable of the ESA project DG-Trac. This document will be used as a reference for implementing security mechanisms in the prototype of the DG-Trac solution that will be developed in the next months. The other partners of the project find the PP to be useful and will rely on it for implementing the security aspects.

There are still several tasks that would be interesting to complete to go beyond the scope of this work and provide new, interesting and advanced approaches for creating PP.

As stated in the previous section, the main limitation of this work is related to the incomplete automation. One interesting future research direction is to try and fully automate the generation of the protection profile. The first step that could be automated can be the definition of the mapping. Using natural language processing, the new approach can try and automatically associate the security objective with the security requirement as defined in the CC document part 2. This technique has already proven to be effective (as shown in [29, 30]) and could lead to some interesting results. It will not be perfect because some mistakes could occur and some security objectives could be linked to the wrong security requirements but if the percentage of errors is low then the approach can be useful.

In addition, the approach needs to be validated by applying it to other case studies having different contexts and requirements. This approach has been applied to one case study, which is not enough to demonstrate that it can scale to handle bigger case studies. For bigger case studies, we need to extend the tool to enable merging sub-trees and handling potential issues when merging trees containing the same security objectives.

Finally, one interesting research direction would be to apply MDE (model-driven engineering) techniques to extend the work by applying model-transformation instead of the simple code-level transformation. There are several benefits gained by performing model-transformation. By creating the meta-models of ADTree and PP, we can perform more advanced analysis on the concepts of the two meta-models, modify their structure or easily perform other model-transformation to other models (like security use-cases and misuse cases).

In a previous work, we have extensively used MDE techniques for access control modeling and testing [31, 32]. These approaches could be applied to ADTrees, which can be very interesting and powerful tool to reason about security threats a system or

a web application could face. We can provide a new methodology for modeling software security based on MDE ADTree.

5 References

1. *Common Criteria for Information Technology Security Evaluation*. [Access 17-12-12]; Available from: http://www.commoncriteriaportal.org/cc/.
2. *USA National Information Assurance Partnership - Common Criteria Evaluation and Validation Scheme for IT Security* [Access 17-12-12]; Available from: http://www.niap-ccevs.org/.
3. *ISO/IEC 15408-1:2009 - Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model*. [Access 17-12-12]; Available from: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50341.
4. *ISO/IEC 15408-2:2008 - Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional components*. [Access 17-12-12]; Available from: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=46414.
5. *ISO/IEC 15408-3:2008 - Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance components*. [Access 17-12-12]; Available from: http://www.iso.org/iso/catalogue_detail.htm?csnumber=46413.
6. Barbara Kordy, Sjouke Mauw, Saša Radomirović, and Patrick Schweitzer, *Foundations of Attack‚ÄìDefense Trees*, in *Formal Aspects of Security and Trust*, P. Degano, S. Etalle, and J. Guttman, Editors. 2011, Springer Berlin Heidelberg. p. 80-95.
7. J.R. Williams and K.M. Ferraiolo, *P3I-protection profile process improvement.* Arca Systems, Inc, 1999.
8. *ISO/IEC ISO/IEC 21827:2008 - Information technology -- Security techniques -- Systems Security Engineering -- Capability Maturity Model® (SSE-CMM®)*. [Access 17-12-12]; Available from: http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=44716.
9. J. Cheng, Y. Goto, S. Morimoto, and D. Horie. *A security engineering environment based on ISO/IEC standards: providing standard, formal, and consistent supports for design, development, operation, and maintenance of secure information systems*. in *Information Security and Assurance, 2008. ISA 2008. International Conference on*. IEEE.
10. D. Horie, S. Morimoto, N. Azimah, Y. Goto, and J. Cheng. *ISEDS: An Information Security Engineering Database System Based on ISO Standards*. in *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*. IEEE.
11. S. Morimoto and J. Cheng. *Patterning Protection Profiles by UML for Security Specifications*. in *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents,*

Web Technologies and Internet Commerce, International Conference on. IEEE.
12. S. Morimoto, D. Horie, and J. Cheng, *A security requirement management database based on ISO/IEC 15408.* Computational Science and Its Applications-ICCSA 2006, 2006: p. 1-10.
13. S. Morimoto, S. Shigematsu, Y. Goto, and J. Cheng. *A security specification verification technique based on the international standard ISO/IEC 15408*. in *Proceedings of the 2006 ACM symposium on Applied computing*. ACM.
14. S. Morimoto, S. Shigematsu, Y. Goto, and J. Cheng. *Formal verification of security specifications with common criteria*. in *Proceedings of the 2007 ACM symposium on Applied computing*. ACM.
15. Paco Hope, McGraw Gary, and Annie I. Antón, *Misuse and Abuse Cases: Getting Past the Positive.* IEEE Security & Privacy Magazine, 2004. 2(3): p. 90 - 92.
16. G. Sindre and A.L. Opdahl, *Eliciting security requirements with misuse cases.* Requirements Engineering, 2005. 10(1): p. 34-44.
17. J. McDermott and C. Fox. *Using abuse case models for security requirements analysis*. in *Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual*. IEEE.
18. D.G. Firesmith, *Security use cases.* Journal of object technology, 2003. 2(3).
19. I. Alexander, *Misuse cases help to elicit non-functional requirements.* Computing and Control Engineering Journal, 2003. 14(1): p. 40-45.
20. G. Sindre and A.L. Opdahl. *Templates for misuse case description*. in *Proceedings of the 7th International Workshop on Requirements Engineering, Foundation for Software Quality (REFSQ'2001), Switzerland*. Citeseer.
21. A.L. Opdahl and G. Sindre, *Experimental comparison of attack trees and misuse cases for security threat identification.* Information and Software Technology, 2009. 51(5): p. 916-932.
22. *SHIELDS Security Vulnerability Repository Service* [Access 17-12-2012]; Available from: https://svrs.shields-project.eu/SVRS/.
23. *ECMA Protection Profile E-COFC Public Business Class*. [Access 17-12-12]; Available from: http://www.ecma-international.org/publications/files/ECMA-TR/TR-078.pdf.
24. *ECMA-205 Standard: Commercially oriented functionality class for security evaluation (COFC)*. [Access 17-12-12]; Available from: http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-205.pdf.
25. *ECMA-271 Standard: Extended Commercially Oriented Functionality Class for Security Evaluation (E-COFC)*. [Access 17-12-12]; Available from: http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-271.pdf.
26. *Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components*. [Access; Available from: http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R4.pdf.
27. *Common Vulnerabilities and Exposures*. [Access 17-12-12]; Available from: http://cve.mitre.org/.

28. *The Attack-Defense Tree Tool, ADTool*. [Access 17-12-12]; Available from: http://satoss.uni.lu/members/piotr/adtool/.
29. Xusheng Xiao, Amit Paradkar, Suresh Thummalapenta, and Tao Xie, *Automated extraction of security policies from natural-language software documents*, in *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering* 2012, ACM: Cary, North Carolina. p. 1-11.
30. Rahul Pandita, Xusheng Xiao, Hao Zhong, Tao Xie, Stephen Oney, and Amit Paradkar, *Inferring method specifications from natural language API descriptions*, in *Proceedings of the 2012 International Conference on Software Engineering* 2012, IEEE Press: Zurich, Switzerland. p. 815-825.
31. B. Morin, T. Mouelhi, F. Fleurey, O. Barais, Y. Le Traon, and J. M. Jezequel, *Security-Driven Model-Based Dynamic Adaptation*, in *25nd IEEE/ACM International Conference on Automated Software Engineering, ASE 2010* 2010.
32. T. Mouelhi, F. Fleurey, B. Baudry, and Y. Le Traon, *A model-based framework for security policy specification, deployment and testing*, in *MODELS 2008* 2008.