UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Salman Lashkarara

Managing Security Risks Using Attack-Defense Trees

Master's Thesis (30 ECTS)

Supervisor(s): Raimundas Matulevicius

Tartu 2017

Managing Security Risks Using Attack-Defense Trees

Abstract:

Nowadays there is an increasing demand for answering the security needs in systematic ways. The In this thesis, we have addressed **risk management using Attack Tree.**

Information System Security Risk Management (ISSRM) is a model which covers all the important concepts in risk management. Also, attack trees are simple and efficient tools for showing the risks. There are few extensions of attack trees, but none of them covers all risk concepts. The said problem limited the usage of attack tree model since it does not consider important measures such as countermeasures, or threat agent's profile.

The contribution to resolve the problem in this thesis includes three steps.

Obtaining an alignment from Attack-Defense trees to ISSRM.

Measurement of the metrics of the nodes of tree using historical data

Implementation of a tool based on obtained tree.

Using the alignment, we have detected the uncovered concepts in Attack-Defense tree. Then we tried to add these concepts to the current Attack-Defense tree. Therefore, the new Attack-Defense tree (called Aligned Attack-Defense tree or A-ADTree) covers most important concepts of ISSRM. In order to measure the risk, we have proposed a mathematical model to evaluate the probability of the nodes in the tree, based on historical data. Then, implemented tool helps to materialize the effect of threat agent's profile, and countermeasures on the risks. The result of implemented tool shows, the obtained A-ADTree has more capabilities (in the evaluation of the probability of risk) in comparison to previous versions.

This solution is capable of giving more hints for the project managers when they are deciding about possible solutions in industries. Additionally, this alignment helps to obtain another alignment between A-ADTree and the other modeling languages in future, since these modeling languages are already aligned to ISSRM.

Keywords:

ISSRM, Attack Tree, Alignment, Risk, Risk treatment

CERCS:T120

Pealkiri eesti keeles

Lühikokkuvõte:

Nagu mujal valdkondades, kasvab tänapäeval vajadus turvalisuse järele nii ka ärimaailmas. Käesolev magistritöö üritab seda probleemi lahendada kasutades riskianalüüsi diagrammi mudelit, mida inglise keeles nimetatakse Attack Tree.

ISSRM (Information System Security Risk Managment) on mudel, mis käsitleb kõiki olulisi riskianalüüsi aspekte, on lihtsalt arusaadav ja annab olukorrast kiire ülevaate. Laiendustena on olemas mõned sellised riskianalüüsi diagrammid, kuid ükski neist pole võimeline käsitlema kõiki võimalikke ohuolukordi. See paneb diagrammi kasutamisele piirid, kuna ei arvesta võimalikke vastumeetmeid ohtudele, ega ohuallika profiili.

Antud magistritöö pakub sellele probleemile kolmeosalist lahendust.

1. luua sild riskianalüüsi puu osast, mis käsitleb kaitsetehnikaid (Attack Defence Tree), kuni ISSRM mudelini;

2. arvestades minevikus ette tulnud riske, riskifaktorite tõenäolisuse ja nendega seotud kulutuste mõõteparameetrite väljatöötamine;
3. tööriista kasutamine, mis on välja töötatud antud riskianalsüüsipuu abil.

Selliselt loodud sild aitab leida veel avastamata aspekte riskianalüüsi puus. Lisades sellise laienduse, on riskianalüüsi puu täielikum ja muudab ISSRM mudeli mitmekülgsemaks. Selleks et riske paremini analüüsida, on kasulik arvestada ka minevikus ette tulnud ohte ning neid matemaatiliselt uurida tõenäolisuse aspektist, et minimeerida sarnaste ohuolukordade taastekkimise tõenäosust. Magistritöö tegemise käigus välja töötatud tööriist (Aligned Attack-Defense Tree or A-ADTree) on võimekam riski tõenäosusele hinnangu andmisel teistest juba olemasolevatest versioonidest.

Antud tööriist annab riskianalüüsi hindajatele rohkem võimalusi võimalike ohuolukordade lahendamiseks ja ennetamiseks. Kuna siin kasutatud modelleerimiskeeled on juba sobitatud ISSRM mudeliga, võimaldab antud töös välja töötatud laiendus luua enam seoseid selle ning teiste modelleerimiskeelte (nt Secure BPMN, Misuse-case diagram, Secure TROPOS, and Mal-Activity diagram) vahel ka tulevikus.

Võtmesõnad:

joondamine, riskide vähendamine, risk

CERCS:T120

Table of Contents

1 Introduction

Nowadays, software is improving performance in the industries. Therefore, the industries and organizations use information systems to support their business. Unfortunately, at the same time, malicious attacks are organized against the information system. Thus, it is crucial to understand the possible risks deeply in the design level of these information systems. It means that the modeling languages should be capable of presenting the risks at the design level.

One of the methods which facilitates recognizing the possible risks is Information System Security Risk Management (ISSRM)[1,2,44]. It also introduces an approach for considering some risk treatment for the identified risk. Additionally, using ISSRM, we are able to evaluate possible risks and risk treatments based on cost. Therefore, it supports decision making in the highest level of management.

ISSRM supports some risk concepts, but since it is not a modeling language, there should be an alignment from ISSRM to the modeling languages. The models like Secure BPMN [4,5,7], Misuse-case diagram [13], Secure TROPOS [9,10], and Mal-Activity diagram [18] have very good features to present risk concepts based on ISSRM.

Attack tree [15] is another modeling language, which presents different possible attacks to an information system. This model may provide some information about the cost of risk, and the probability of it. Although the attack trees are a common tool to present a different kind of risks on a system, they are incapable of representing some important aspects of information systems. In fact, the attack trees simply mention the different approaches to achieve a malicious goal without pointing to the assets, vulnerabilities, and security requirements. However, the attack trees are useful, they do not provide enough information for the software engineers to avoid the probable risks. There are different extensions of attack trees such as vulnerability tree, or protection tree. One of the extensions is Attack-Defense tree [25,26], which may show both risk concepts and risk treatment in a tree structure. We consider the ISSRM as a complete reference which shows all the essential concepts of information systems. Therefore, there is a need to make an alignment of the Attack-Defense tree to ISSRM.

However, the Attack-Defense tree has the potentiality of presenting the risk, and risk treatment concepts, it cannot show the vulnerability or the asset in an information system. Additionally, it is important to evaluate the cost and probability of nodes in the tree properly.

One of the methods to measure the probability is OCTAVE [37]. This method tries to evaluate the probability of attack nodes mostly subjectively and based on the opinion of experts. Although this feature simplifies the estimation of probability, it is very subjective and could lead to different results in similar cases. Also, the evaluation of OCTAVE is qualtivative.We have tried to propose a statistical method based on OCTAVE which evaluates the probability of the risk based on historical data.

Main Q. How can we align attack tree to Information System Security Risk Management?

The main question is decomposed to three questions?

Q1. How can we show the main elements of risks in information systems?

Q1.1.What is the ISSRM?

Q1.2. What are the existing alignments of modeling languages to ISSRM model?

Q1.3.What is attack tree?

Q1.4. What is the meaning of multiset semantic?

Q1.5. What are the concert and abstract syntax of an attack tree?

Q1.6. What are the main measurement approaches in attack trees?

Q2.How can we align Attack-Defense tree to ISSRM?

Q2.1.What Attack-Defense tree constructs could be aligned to the ISSRM domain model?

Q2.2 What is the evaluation of metrics in both attack, and defense nodes?

Q2.3 What is the new measurement model in aligned Attack-Defense tree?

Q3. How can we validate the aligned Attack-Defense tree?

Q3.1 What is design for implementation of aligned Attack-Defense tree?

Q3.2 What are the examples of risk mitigation using aligned Attack-Defense tree?

In this thesis, we have tried to make an alignment from Attack-Defense tree to ISSRM. The alignment helped us, to detect the missed concepts of ISSRM in Attack-Defense tree. Then, we have added the missed concepts to the Attack-Defense tree, and we have obtained a new Attack-Defense tree (A-ADTree), which shows important the concepts of ISSRM. In the next step, we have implemented a tool based on the derived Attack-Defense tree. In order to obtain the probability of risk, we have proposed a mathematical model which works based on historical data, OCTAVE, and Bayes theorem. At the end, we have implemented a tool to show how the A-ADTree works.

The second chapter includes a literature review over ISSRM and the other modeling languages alignment to ISSRM. Chapter three is about attack tree families, and the measurement approaches. In chapter four, we explain the alignment of Attack-Defense tree to ISSRM. In chapter five, we explain the structure of historical data and evaluation of probability. The sixth chapter explains the risk measurement in the aligned Attack-Defense tree. Chapter seventh, explains the design, and requirement of implemented aligned attack-defense tree. And the last chapter provides two examples to show how A-ADTree (aligned Attack-Defense tree) works.

Part I: State of the Art

2. Security Risk Management

Information System Security Risk Management [1,2,44] is a conceptual model which shows the main concepts which are involved in each risk. It contains both risk, risk treatment, and the assets in an information system. We consider ISSRM as a reference for risk management concepts.

Also, language models facilitate the design of the whole system. It is important to show the risks at the design level. Using alignment, we detect the concepts of ISSRM to in a system. There are some well-known models such as BPMN [4,5,7], Misuse case diagram [13], Mal-activity diagram [18], and secure TROPOS [8,9], which are aligned to the ISSRM.

In this chapter, we answer

Q1.1.What is the ISSRM?

Q1.2. What are the existing alignments of modeling languages to ISSRM model?

2.1 ISSRM

Information system security risk management is a domain model which presents the main items in an information system from the perspective of security [1]. These items include (1) **Assets concepts**, (2) **Risk concepts**, and (3) **Risk treatments concepts**. Fig 1. shows the domain model of ISSRM.

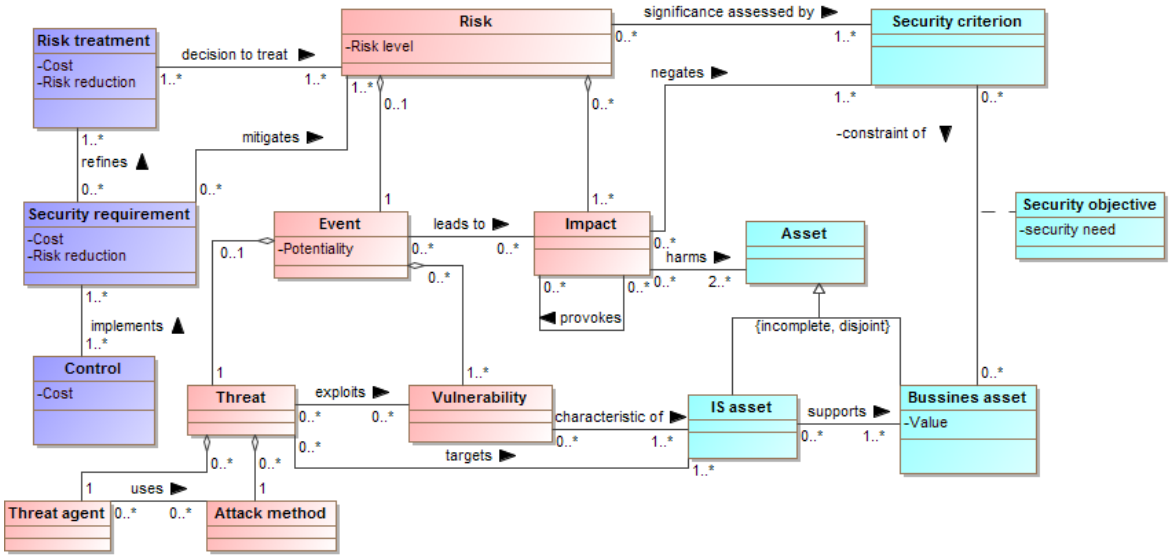


Fig 1. The ISSRM Domain model- adapted from [1][2]

Assets concepts: In ISSRM, the assets mentions to all valuable things in the organization. It has two main categories, Business assets and Information System assets (IS Asset). **Business Assets** are the assets which are related to the processes, business, information, or essential skills of the organization. IS Asset consist of all the valuable things related to the information system part. Therefore, the equipment like database, networks, routers and operating systems are among IS assets. All assets can be a target of malicious actions. Thus, it is essential to consider the **security criterion** for them. There is three main criterion in this part, integrity, availability, and confidentiality.

Risk concepts: In the ISSRM a risk is a combination of several different concepts. The most important ones are threat agent, attack method, vulnerability, and impact. The rest of concepts (like Event, Threat, and Risk) are derived from the aggregation of these four concepts. A threat agent is anyone that has skills and motivation to do a malicious attack. An Attack method is a process which may lead to a problem in the whole system. A threat is potential possibility that, an attacker uses an attack method against an asset. Also, a threat exploits some vulnerabilities in the information system assets. These vulnerabilities are characteristics in the IS assets which facilitate the threats.

In case a threat agent uses some attack methods to exploit the vulnerability against one or more IS assets, then we face with an event which leads to some negative impacts on the system. These impacts are a negation of security criterion. They harm both IS or business assets. Finally, a risk is an aggregation of negative events and their impacts.

Threat Agent: Threat agent is one of the key concepts in the ISSRM. A threat agent can be described as **motive, opportunity, capabilities,** and **means** [36]. Motive discusses the source of motivation of the threat agent. Means point to the equipment which attacker may need to perform the attack. Opportunity mentions to the time of attacker to perform his attack and make several mistakes. Capability states the knowledge and skills of the attacker.

Risk Treatment Concepts: Risk-treatment concepts involves three main factors. The first factor is *risk treatment* which describes the policy to face the risk. This policy could be one of the risk avoidance, risk reduction, risk transfer, or risk retention. The risk treatment part is a management decision part. The second factor is security requirement, and the third one is controlling. Security requirement gets extracted based on the risk and risk treatment policy. The control is an implementation of security requirements. The cost attribute helps to evaluate the solutions from the perspective of the economy.

Metrics: The ISSRM security metric is introduced in [3]. It tries to evaluate the effectiveness of security controls based on risk level before and after applying the risk treatment solutions. The risk level simply describes, the possibility of risk based on the potentiality of the event and its impact. Also, potentiality depends on the likelihood of threat and vulnerability level of an IS asset. The threat likelihood and vulnerability level are the qualitative measures which are mapped to a set of numerical value.

2.2 Overview of Modelling Languages

BPMN: Business Process Management (BPM) is the science to manage processes in an organization. One of the purposes of BPM is to find defects in the process and mitigate them. Such improvements benefit both customers and organization. The Business Process Management and Notation (BPMN) is a process modeling language which presents how a process works in an organization in detail [4][5]. In [7] the authors introduced an extension of BPMN with respect to the ISSRM. This extension supports the presentation of three main components of ISSRM using an alignment. Using this modeling language, it would be easy to consider security requirements, security risk, and endangered assets during the early steps of system design.

TROPOS: In [8], Bresciani introduced an agent-oriented model to develop software. This model has few steps. Firstly, the problem and the organization get defined. Secondly, the functional requirements get documented. Then, the relation between the main components (architecture), and their interaction should define. Finally, the implementation of the system happens. Secure TROPOS is an extension of this model [9][10][11].Its main point is to integrate security during the development process based on TROPOS method. It is important to find a way to consider security during the design and early steps of development.

Although TROPOS shows the security requirements with soft-goal, it presents soft-goal in the third step (architecture). Therefore, the security is not considered in the first two steps of design (early and late requirements). Additionally, it is unable to present some security constraint in the system. The objective of secure TROPOS is to cover these problems.

Misuse Case: Misuses case [13] is an extension of the usecase diagram to present risk related issues in an organization. The usecase diagram can present functional requirements well, but it is not able to present non-functional requirements like security. Misuse case tries to answer to such needs in the usecase diagram. It simplifies understanding of security issues. Then it would be easy to define a set of requirements based on a defined security issue. However, Misuse case is easy to use and learn, its output is hard to analyze [6].

Mal-Activity: Mal-Activity diagram is another modeling language to present possible negative activities in an information system [18]. An activity diagram is used to show a scenario or a usecase in an organization [19][20]. To draw a mal-activity, first, we draw an activity diagram to depict normal procedure, then add the mal-activity. Like Misuse-case diagram, the main purpose of this diagram is also, presenting the illegitimate and legitimate activities at the same time. Although the Mal-activity and Misuse case diagrams follow the same approach, they are complementary in some aspects [6]. Table 1. shows the concrete syntaxes in each languages which are used to show ISSRM concepts.

Table 1. Alignment of different languages to **asset** concepts- adapted from [2]

ISSRM	BPMN	Secure TROPOS	Misuse Case	Mal-activity
Assets	Event, Task, Gateway, Sequence flow	Actor, Hardgoal, Plan, Resource, different dependencies, means-end, decomposition links, contribution	Actor, Usecase, extend, and include links	Decision, Activity, Control flow
Business Asset	Data Object			
IS Asset	Data Store, Pool		package	Swimlane
Security criterion	Lock (Confidentiality, Integrity, Availability)	Security constraint, Softgoal, Decomposition links	Security criterion	--

2.3 Language alignments to ISSRM domain model

In [12] the author explains the alignments of the modeling languages to the ISSRM. The alignment shows presentation of the three main components of ISSRM with the constructs of each language. Alignment shows which constructs have the potentiality to match with which class in the ISSRM domain model. In [14] the author summarized the alignments as it is presented in Table 5, Table 6, and Table 7. The alignment of some languages is not so precise. To deal with such problem, it is useful to define and add a new construct to the language to fulfill the requirements. For instance, in [7], the authors introduced a new concept (called lock), to present the security criterion in asset concepts of ISSRM in BPMN. Also, vulnerability point used to show the vulnerability in Risk concepts in the same language. The idea of vulnerability point also was used in Secure Tropos too [34]. Risk treatment does not have any alignment in any languages because it is a policy about how to deal

with risk. The alignment of security requirements presents the risk treatment and control at the same time. Table 2, and 3. present the summary of alignments to risk, and risk treatment concepts in different languages.

Table 2. Alignment of different languages to **risk treatment** concepts- adapted from [14]

ISSRM	BPMN	Secure Tropos	Misuse Case	Mal-activity
Risk treatment	-	-	-	-
Security requirement	Task, Event, Gateway, Sequence flow	Actor, Hardgoal, Plan, Resouce, Security constraint, dependency, contribution,means-end, decomposition link	Security usecase, Extend and include link	Decision, mitigation activity, control flow
Control	-	-	-	Mal-swimlane

Table 3. Alignment of different languages to **risk** concepts- adapted from [14]

ISSRM	BPMN	Secure Tropos	Misuse Case	Mal-activity
Risk	Aggregation of Event and Impact	Aggregation of Event and Impact	Aggregation of Event and Impact	Aggregation of Event and Impact
Impact	Lock	Impacts arrow	Usecase with Impacts stereotype	Mal-activity Swimlane
Event	Aggregation of vulnerability and Threat	Threat or, Aggregation of vulnerability and Threat	Aggregation of vulnerability and Threat	Aggregation of vulnerability and Threat
Vulnerability	Vulnerability point	Vulnerability point	Usecase with Vulnerability stereotype	
Threat	Aggregation of Attack method and Threat agent	Goal, Plan	Aggregation of Attack method and Threat agent	Aggregation of Attack method and Threat agent
Attack method	Task, Event. Gateway, Sequence flows	Plan, Task, Decomposition link	Misuse case, include and exclude links	Mal-Decision, Mal-Activity, control flow links, Mal swimlane
Threat agent	Pool	Actor	Misuser	Mal-swimlane

2.3 Summary

Fig 2. presents the process of extracting risks and risks treatments. It starts with the identification of important assets. Then the security objectives of this assets must be recognized. The security objective of each asset comes from the asset's role in the final product or service. In the next step, risk analysis part should get performed. The last three parts are related to the risk treatment. After identification of possible risks, we make a decision about the general approach to face with risk. Then, some security requirements get defined based on risk treatment. At the end, a set of controls are implemented. Since there is no guarantee

for the efficiency of the whole process, this process should be done iteratively, and the risks must get monitored always.

In [14] the author classifies each class in ISSRM based on the principle of semiotic clarity[22][23]. Semiotic clarity discusses the preciseness of alignment between two different languages. If two constructs from two different languages provide exactly the same meaning then there is a one to one relationship between these two elements, otherwise, it could be overload, incompleteness, redundancy, or under-definition. Redundancy means in the referent language, there is two or more constructs witch conveys the same meaning. Overload mention one given construct can be used in different alignment. Incompleteness means, there is no semantical equal construct in the referent language. Also, under construct refers to a situation when a construct has not special equivalent semantic. Table 5. shows the comparison of alignment of them to the ISSRM with respect to semiotic clarity.

The current language models do not cover the main concepts of the security needed to secure information system. Table 4. shows the three main concepts of ISSRM covered by the other models. Table 4. justifies the alignment of these languages to ISSRM. The alignment helps that in case that the model does not cover a concept, the missed concept get compensated with extra elements.

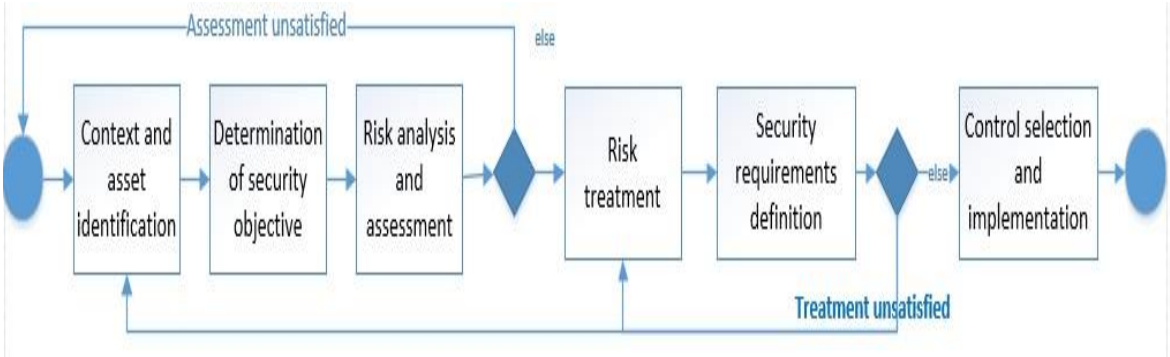


Fig 2. The process of ISSRM, adapted from [2]

Table 4. Summary of covered concepts in different models **before** alignment

	Asset	**Risk**	**Risk treatment**
BPMN	✓		
Secure Tropos	✓	✓	✓
Misuse case diagram		✓	✓
Mal-activity diagram	✓	✓	

Table 5. Comparing Modelling languages in alignments of ISSRM - adapted from [14]

Semiotic clarity	**BPMN**	**Secure Tropos**	**Misuse cases**	**Mal-activity**
One-to-one correspondence	Threat agent	Threat agent	Security criterion, Vulnerability, Threat agent	Impact, Threat agent, Control
Redundancy	Assets	Event	Assets	Assets, Attack method
Overload	Assets	Assets	Assets	Assets
Incompleteness	Security criterion, Risk, Impact, Event, Vulnerability, Threat, Risk treatment and control	Risk, Impact, Event, Vulnerability, Threat, Risk treatment and control	Risk, Event, Threat, Risk treatment and control	Security criterion, Risk, Event, Vulnerability, Threat, Risk treatment
Under-definition(excess)	Assets, Attack method, security requirement	Asset, Security criterion, Attack method, security requirement	Asset, Attack method, Security requirement	Asset, Security requirement

3 Attack Tree

In this chapter, we answer

Q1.3.What is attack tree?

Q1.4. What is the meaning of multiset semantic?

Q1.5..What are the concert and abstract syntax of an attack tree?

Q1.6. What are the main measurement approaches in attack trees?

An attack tree [15] is a top-down technique that helps to present all possible attack methods to a system. This model is tree [16] data structure. Every node in this tree can have several nodes. Each node can be decomposed to children nodes, and children nodes can also have their own children. Therefore the parent nodes are goals and the children nodes are the ways to achieve the parent. The decomposition structure continues till the leaf node shows a single attack. The relation between the children of a node could be **Conjunctive** or **Disjunctive**. Also, these relations sometimes called refinements. The Conjunctive relation means that the parent can be satisfied only when all the children are achieved. Meanwhile, the disjunctive means that the parent node could be satisfied if one of its children get achieved. There are many different types of attack trees like Vulnerabilities Tree [17], Protection Tree [21], Defense Tree [31], and Attack-Defense trees [26][27]. All of them have the above features of attack tree. There are only a few things which distinguish them.

In [38] the author suggests a semantic for attack trees. Based on this semantic, the structure of the attack trees does not affect the interpretation of attack trees. They consider the atomic attacks as leaves of the tree. These atomic attacks have Boolean values based on success or failure of the attack. The rest of the nodes contain Boolean operators (AND, OR). The value of the root in this tree is the output of the function. Mauw and Oostdijk suggest the attack tree could have a different structure, but the final output of them should be the same.

Additionally, in [38] the authors provide two reduction rules which keep the semantic of attack tree but decrease the complexity of it. At the first step, they suggest that a node is connected to a multi-set of nodes with an edge. They called this multi-set as a bundle. The rules are as following:

1) If in a bundle, a node has only one connection to a bundle, then this node could be removed and the child bundle replaces one level up.
2) If a bundle contains a node which has a connection with two or more bundles, then this bundle could be replaced with two bundles. The first one contains the first bundle, and the second one contains the second bundle.

3.1 Multiset Semantic

As it discussed before, an attack tree explains the different approaches which an attacker may use to achieve a given goal. Some of the steps are atomic, which means they are not decomposable to smaller attacks. On the other hand, some attacks are composed of smaller attacks. These attacks are made by smaller attacks. The atomic attacks may happen multi times in an attack tree. As a result, a component attack could be considered as multisets of atomic attacks.

Definition: Assume the collection of component attacks is called C. Then and attack is a finite, none empty, multiset of attacks. Attack suit is a finite set of attacks.

Example: Assume the set C to achieve administrator passwords is {bribe, force, install a key logger, access computer, find administrator}. Then the attack for this set of components is { find the administrator, bribe}, {find administrator, force},{access computer, install key logger}. The attack suit is the set of all three above sets:

{{find administrator, bribe}, {find administrator, force},{access computer, install key logger}}

3.2 Family of Attack Tree Languages

Defense Tree is an extension of attack tree, but in there are some countermeasures added to the leaf nodes of attack tree. This tree should be presented in disjunctive normal form (DNF). In this format, the AND nodes are the leaves of the tree. Such format simplifies the whole tree.

Vulnerability Tree has a vulnerability in its root. The children are the events which could lead to exploiting this vulnerability. There are AND, and OR connectors to connect the nodes together. The connectors perform the refinements roles.

Protection Tree is a complementary tree to avoid attacks based on attack tree. In this approach, firstly, the attack tree get designed. Then the protection tree is developed. In protection tree, there is a given node with respect to a certain node in the attack tree. The nodes in the protection tree present a solution to avoid the attack in the corresponded node in the attack tree.

Attack-Defense tree can be considered as a zero-sum game with two players in game theory. In ADTree, there are two different nodes. The opponent (attack nodes) and the proponent (defense nodes). Every node regardless of its type can get refined to some children. The leaf nodes perform a basic action and do not refine. Additionally, every node can have only one child of the other type, which depicts a countermeasure for that node. The definition of Conjunctive or Disjunctive operators is alike the original attack tree. ADTree provides a countermeasure for both types of the nodes. Such feature helps to develop a possible attack tree to more details. Table 6. shows the ADTerm respected to each ADTree.

In [28], the authors pointed that an Attack-Defense tree could be presented using a mathematical syntax called ADTerm. The below table defines the meaning of notations in ADTerms. The S={Attack node(p), Countermeasure node(o)}

Example: For an ADTrem like $c^{\boldsymbol{p}}$ ($\wedge^{\boldsymbol{p}}$ (a,b),$c^{\boldsymbol{o}}$ (d,e)) the equivalent ADTree would be fig 3.

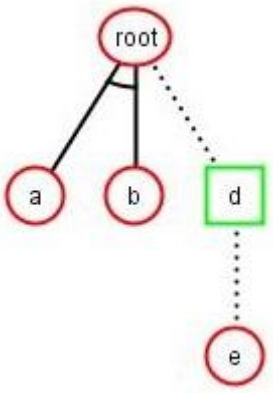
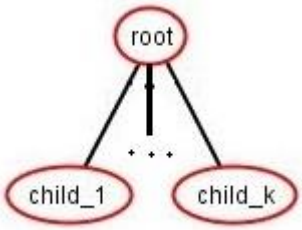
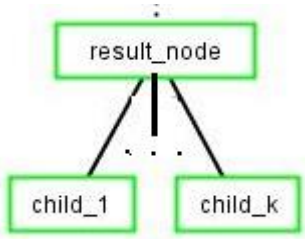
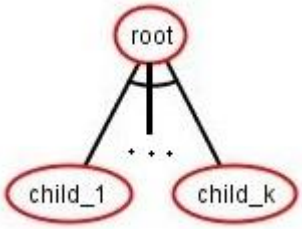
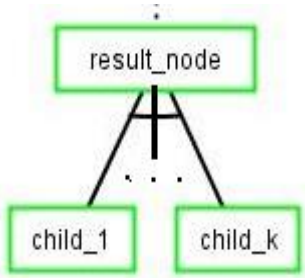
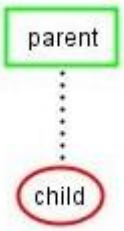
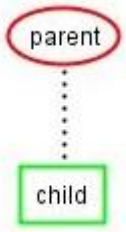


Fig 3. ADTree corresponds to the $c^{\boldsymbol{p}}$ ($\wedge^{\boldsymbol{p}}$ (a,b),$c^{\boldsymbol{o}}$ (d,e))

Attack Graph [32] is also a technique for analyzing of network security vulnerability. A path in this graph present all the essential actions which attacker needs to perform to intrude the network. In this graph, the nodes present the state of the network during the attack. There are three main factors which help to generate the attack graph. These factors are Attack template, a detailed description of the system, and the attacker's profile.

Table 6-Mathematical, and Graphical representation of formalized notation

Mathematic expression	**Graphical meaning**	
$\vee^s$ There are disjunction refinement between children of a node		
$\wedge^s$ There are conjunction refinement between children of a node		
c^s A countermeasure is applied on a node	c^o 	c^p 

Semantic Threat Graph [33] is another technique to risk analysis. Although the attack trees are good in presenting high-level attacks, they have some weaknesses. Attack trees are not able to present the detailed security solutions. For instance, based on ADTree, a possible countermeasure for Denial of Service attack, would be installing a firewall. However, the configuration to deal with different kind of DoS attacks can be different. Additionally, the Semantic Threat Graph includes the other important concepts. There are four main concepts in the STG: ***threat*** (which basically covers the potentiality for attack), ***asset*** (which is the final purpose of the threat, and could be either people in the organization or infrastructure), ***vulnerability*** (which is a weaknesses in the assets), and ***countermeasure*** (which is an action or a process to mitigate the vulnerability and prevent the threat). These concepts are connected to each other using directed edge. In fact, the STG is a directed graph and not a tree.

Among all different extensions of attack tree and attack graph, the ADTree is more suit for the purpose of this thesis. It has the potential to present both the risk and risk-treatment

concept at the same time. Although the defense tree could also fulfill some needs, it can only show the countermeasures against attacks in leaves. Moreover, STG has the potentiality to present the ISSRM, but for the purpose of this thesis, we are interested in the trees. Additionally, in [25], the authors provide a free graphical open source tool to support the AD-Tree which is called ADTool. This tool facilitates the third part of the thesis.

Table 7. Comparing family trees based on ISSRM concepts coverage.

	Asset	Risk	Risk-treatment	Tree support
Attack tree	No	Yes	No	Yes
Defense Tree	No	Yes	Yes (only in leaves)	Yes
Vulnerability Tree	No	Yes	No	Yes
Protection Tree	No	No	Yes	Yes
Attack-Defense tree	No	Yes	Yes	Yes
Attack Graph	No	Yes	No	No
Semantic Threat Graph	Yes	Yes	Yes	No

3.3 Attack Tree-Defense Tree

The elements in the nodes can have both numerical and textual values. It facilitates both technical and non-technical analysis [24]. A node can have different elements based on the final purpose of analysis [15]. For instance, cost element provides some information regarding the expenses of the attack for the attacker. Also, the node may provide some elements regarding the skills level of attacker, needed time of the attack, or the special equipment which an attacker needs to perform the attack. Another possible attribute could be the likelihood of the attack. In [24], the author also used impact as a qualitative measure and map them to a set of numeric ranges. Additionally, the risk is included and evaluated based on the other element. The final evaluation of risk can be done either by Bayesian network methods [29], or some simpler approaches [3].In [30], the author suggests a very simple bottom-up approach make a quantification analysis on an attack tree. In this approach first, the leaf nodes get their values and then the whole tree get traversed and based on the refinements between the nodes the parents get values. All of these elements facilitate analysis to detect the most probable attack based on some metrics. Table 7. compares the capability of the ISSRM coverage on different attack tree.

3.4 Concrete Syntax

In [25], the authors provide an open source tool for graphical representation of Attack-Defense trees. Fig 4 presents the concrete syntax of Attack-Defense tree. Although, the root node can be either defender or attacker, in this thesis we consider the root node as the attack node. In this tool, the only elements which a node has are its label. The label has a textual value that describes the objective of that node. We can consider the Attack-Defense tree as a game between two players [27].Moreover, it is supposed that the nodes of each player are a countermeasure for the other player. The refinements between two nodes from the same player would be a simple line. Meanwhile, the line between a node and its countermeasure is a cut line. Every node can have an unlimited number of children from its kind, but it can only have one countermeasure node. Such policy helps to keep the tree simpler [28]. By default the operation between children of a node is conjunctive, but it can easily get changed to disjunctive too.

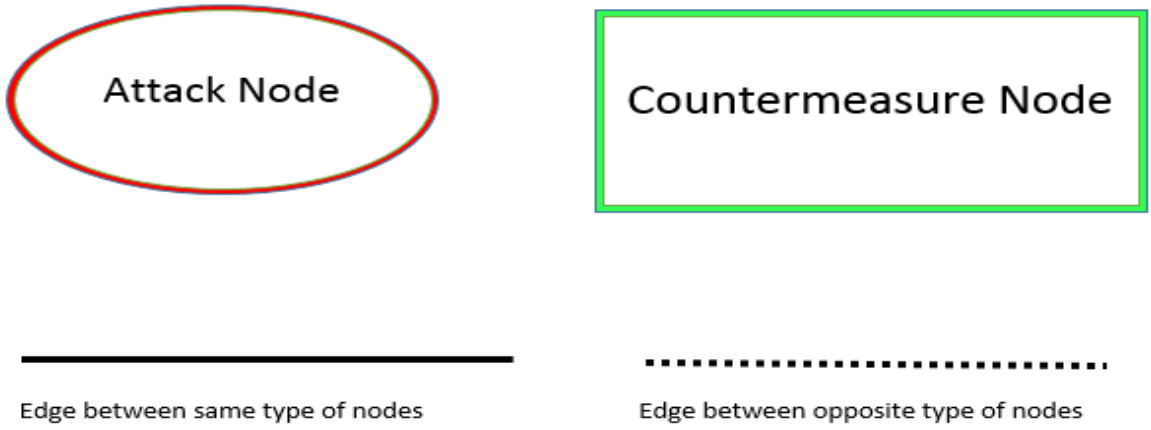


Fig 4. The concrete syntax of Attack-Defense tree.

3.5 Abstract Syntax

Fig. 5 shows the relationship between different graphical elements of Attack-Defense tree. Both attack node and countermeasure node inherit the same feature from the abstract class of node class. An instance of Node could be either Attack Node or Defense Node. The multiplicity of Node class shows that an instance of this class may have only one cut line, but many line instances. The generalization of the cut line and Line from Edge class shows that a Node may come from either a cut line or a Line instance. Therefore, every node has only one parent. The attribute To_Countermeasure_Child states that every node can have only one countermeasure. Meanwhile, the To_Child shows one to many relations between a node and Line class.

3.6 Measurement Models

The nodes in attack tree need to get evaluated in order to provide information for the managers who decide about the security of the information systems. Usually, managers look for the optimal level of security or return of security investment (ROSI). In order to achieve this goal the attack nodes should have some attributes with some quantitative values. These quantitative values could be the likelihood of attack, and the cost of performing of it. The process of quantification of attack tree is called attack tree analysis. There are two main approaches for this purpose.

In one approach, first attributes of the leaves get evaluated. Then in a bottom-up approach, based on the values of leaves, the other nodes of the attack tree get computed. In this approach, the conjunctive nodes mean that all of the children nodes should get evaluated, but the order of execution of nodes and possibility of a repetition of failure nodes is not important.

Alternatively, analysis of an attack tree could be considered as a Boolean function. An attack tree like A can be refined to several simpler attacks like $A_1, A_2, \dots, A_m$. The two main refinements are:

"**AND**" ,also called conjunction, refinement A=$A_1 \wedge A_2 \wedge \dots \wedge A_m$ means to materialize the A all the children $A_1, A_2, \dots, A_m$ should happen.

"**OR**" ,also called disjunction refinement A=$A_1 \vee A_2 \vee \dots \vee A_m$ means to materialize the A at least one of the children $A_1, A_2, \dots, A_m$ should happen.

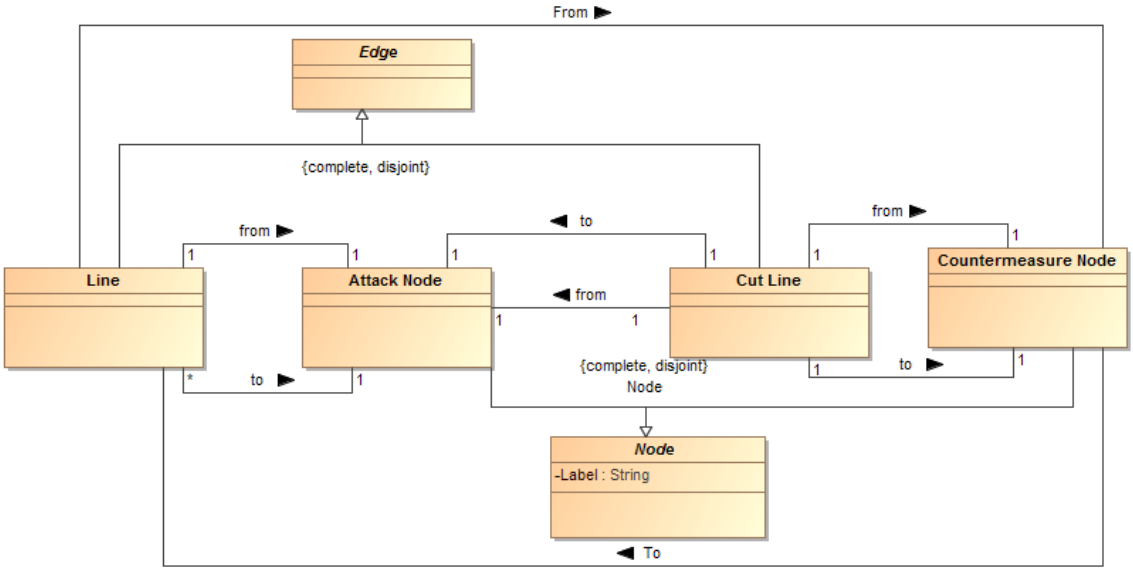


Fig 5. The abstract syntax of Attack-Defense tree.

The whole attack tree can be shown as recursive function with some nodes which none of the refinements can apply to them ($X_1, X_2, \dots, X_m$). These are the leaves node that called **atomic nodes**. Therefore, the attack tree is A can be presented with a Boolean monotone function like A($X_1, X_2, \dots, X_m$). The function gets a set of steps (atomic attack nodes in leaves) and the output of the function is the result of analysis (1 the attacker win, or 0 attacker lost the game).For a given node like X_i the values could be true, false, or $\perp$ ($means\ that\ the\ attacker\ get\ arrested\ and\ should\ pay\ the\ penalty$). This means the given set of steps can lead to a successful attack at the end or not. These steps are called **attack suites**. The other concept in this approach is **attack strategy**, which decides the next step of the attacker based on the previous results. In other word, attack strategy clarifies if the attacker can repeat a failure attack, the order of the chosen steps, and the condition to leave the attack scenario.

3.6.1 Multi-Parameter Attack Tree Analysis

In [39] the authors suggested the rational financial reasoning for the attackers as following:

1) The attacker attacks the system only if the value of the attack overweight the costs of it.
2) The attacker always chooses the optimal way to attack.

Based on these two rules, we can decide an information system target is safe only if the attacker does not achieve benefit from attacking the system.

Although, many people tried this approach with only one parameter at the time, in [39] the authors suggest multi-parameter attack trees, which benefits from game theory and rational economic reasoning to perform the attack tree analysis. Using this method, it is possible to estimate cost and probability of success, then using the rational reasoning and game theory, we can decide if the information system really is attacked or not.

They consider that the chance of a successful attack in an attack node is **p**, and the chance of getting arrested after a successful attack is **q**. The attacker pays a certain **Cost** to perform the attack. In case the attacker is arrested he should pay a penalty. The penalty of a successful attack is $\boldsymbol{\pi}$. It is also possible that the attacker is arrested after a not successful attack. This

probability is mentioned as $q_{\_}$. In this case his penalty would be $\pi_{\_}$ .In case the attacker successfully perform the attack, he achieves some **Gain**. Table 8. summarizes the whole possibilities with their probabilities. The probabilities of **q** and $q_{\_}$ and the value of $\pi_{\_}$ and $\pi_{\_}$ is not necessarily the same. Therefore, in the leaf node there are four parameters to evaluate (Cost, p, π, $\pi_{\_}$).

Table 8. summary of attack actions and the outcome of them

Attack Success/Probability	Attacker get arrested/Probability	Final outcome
Yes /p	No/1-q	P*Gains-Cost
Yes/p	Yes/q	P*Gains-Cost- π
No/1-p	No/1-$q_{\_}$	Cost
No/1-p	Yes/$q_{\_}$	-Cost-$\pi_{\_}$

The general formulae to calculate the final outcome of the attack is:

Outcome=-Cost+p*(Gains- π)-(1-p)* $\pi_{\_}$

The authors considered the atomic attacks in leaves as independent attacks. Then, they provided formulas for computing the parameters for AND nodes and OR nodes.The presented attack tree analysis method has three main shortcomings:

1) It considers the attack leaves as independent events. In the real world, the atomic attacks may not be independent. It makes the problem when there is an AND node in the tree. In this case, the values of one child may propagate to parents more than once. Fig 6. shows a dependent tree.
2) This model is not compatible with the semantic provided in [38]
3) The notation of attacker behavior, preferences, and his strategy is not considered in this model. This model does not provide any information about the order of attack node to get performed. Additionally, it does not clarify when the attacker stops the game.

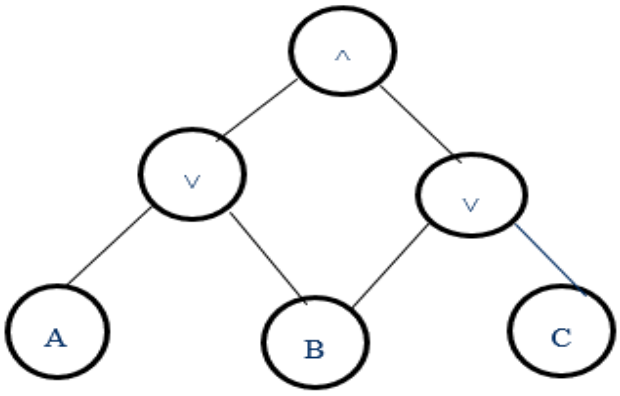


Fig 6. Dependent Tree

In part 2.2, we have mentioned that a threat agent in risk concepts in ISSRM is described with his ***motive***, ***capabilities***, ***opportunities***, and ***means.*** Except for the motive, the rest of factors affects the p in the parallel model, meanwhile the π, $\pi_{\_}$, **q**, $q_{\_}$ are social elements and are not affected by the feature of threat agent. Table 9.shows the relationship between attack parameters in [39] and threat agent attributes in ISSRM

Table 9. correlation between attack parameters in [39] and ISSRM threat agent

Attack Parameters in [39]	Threat agent parameters in ISSRM
P (possibility of success in an attack)	Capabilities, Opportunities, Means
π, π_{-},q , q_{-}	--

3.6.2 Parallel Model

In [40] the authors presented a new analysis model to compensate the shortcomings of [39]. This method is also base on monotone function. The following is the procedure to perform it:

- Create the attack tree and evaluate the value of leaves with the four parameters in [39].
- The attacker finds all the potential attack suites. Some of these attack suites can lead to a positive result in the root. These attack suites should be evaluated.
- The attacker computes the outcome of all the attack suites which lead to a true result in the root. Then, he chooses the one with the highest outcome.

Since this method works with Boolean function, therefore it does not depend on the structure of the tree in the analysis. As a result, it fixes the shortcoming of [39] related to the inconsistency with semantic of attack tree. It also solves the problem related to the dependency of attack steps. Although this model resolves two major problems, it still suffers from some issues. The attacker strategy mentions that the attacker performs the whole attack suites, but he has only one trial which is a paradox. Also, in case the attack tree is big, performing all the attack suites increase the complexity of analysis.

3.6.3 Serial Model

Serial model [41] is an extension of the parallel model. In this model, the attacker tries to smartly not to perform all the leaves which have not any effect on the final result of the Boolean function. The attacker is not allowed to repeat a failure attack, but he can choose the order of his attacks in advance, but later he can refuse to perform some of them. Here is the complete strategy of the attacker in this model.

1) Create an attack tree with a set of leaves $\boldsymbol{X} = \{\boldsymbol{X_i}: \boldsymbol{i} = \boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{n}\}$.
2) Consider a subtree σ of X ($\sigma \subseteq$ X) which materialize the root value as true.
3) Consider a permutation α of σ.
4) Compute the outcome considering the σ and the α.
5) Find the maximum outcome among all the choices for σ and the α.

Although, this model tries to skip some of the paths in the attack tree it still suffers from complexity because all the combination of leaves which leads to materializing the root should be evaluated. Additionally, however, the model provides more information for attacker strategy it still is not compatible with the attacks in daily life. It may happen that the attacker tries some attacks and based on their results chooses the order or combination of attack steps. Also, in some cases, the attacker should follow an exact order of attacks which is not changeable.

3.6.4 Fully Adaptive Model

The authors of [42] present a new model to fix the complexity issue in the previous model. They suggested computing the upper bound of the outcome. If the upper bound of outcome is negative, then it means the expenses of attack are more than the costs, therefore the system is secure. Also, the article suggests recursive formulae compute the exact outcome. The model resolves the complexity problem, but the attacker strategy still has some problems. The model suggests that the attacker can only perform every attack once, meanwhile it is not true in real world. Also, in case the attacker get arrested, then the attacker would be game over, meanwhile the attacker can continue to attack even after arresting. For a Boolean monotone function like F($X_1, X_2, \dots, X_m$) the x_j is an input variable corresponds to the atomic attack x_j the exact utility at node j is U_j

$$U_j = -C_j - \left(1 - p_j - q_j\right) * \pi_j + q_j * U_{(A_{x_j}=0)} + p_j * U_{(A_{x_j}=1)}$$

The authors showed that in case of And refinement between atomic nodes the optimal choice to maximize the outcome is $\frac{C_i}{1-p_i}$, and in case of Or refinement $\frac{U_{(X_i)}}{1-q_i}$

The fully adaptive model can be used when an atomic attack is repeated in the other subtrees. If there is a repeatable attack, its cost should be reduced to the whole numbers of attack. Here is the algorithm to which works with uniform cost reduction.

1) Reduce the cost of repeatable actions to $\frac{C}{1-q}$
2) Sort all the new nodes by $\frac{C}{1-p}$. The node with the lowest ratio should be the first node in subtree.
3) Compute U(A) for each subtree.

Table 10-summary of problems of monotone Boolean function methods

Model\Issue	Independent nodes	Compatibility with [38]	Attacker Strategy	Complexity	One trial
Multi-parameter	✓	✓	✓		✓
Parallel			✓	✓	✓
Serial			✓	✓	✓
Fully Adaptive			✓		✓

3.6.5 Vulnerability Tree Analysis

In [17], the authors also suggested a solution to analyses the vulnerability tree. Based on this approach, the damage analysis table is a table which its rows show the attacks in the leaves and the columns contain the analysis. The first column is the list of damages, and later this list will be categorized to fewer classes of damages. The next column contains the evaluation of each category in the first column. The third column in damage table is the probability of each attack based on the previous history. The fourth column is the financial risk value which can be calculated using the probability of occurrence of the attack multiply by the value of the damage. The final column is the normalized damage which is a value between 1 to 100.

3.6.6 Protection Tree Analysis

In [21], every node in the attack tree node has four attributes. The attributes are ***risk***, ***impact***, ***cost***, and ***probability*** of success. The impact is the effect of the attack on the system which is a value from 1 to 10 and is extracted from a given table which describes the effect of attacks in the system for a certain range of numbers. Probability and cost are derived from historical data. The article used a bottom-up approach to evaluating the other nodes in the tree. They have considered the nodes in the tree as an independent node.

3.6.7 Defense Tree Analysis

In [31], the authors suggested an economic framework evaluate the effect of adding countermeasures to the leaves of attack tree. The metric to measure the economic gain of the attacker is Return of Attack, which is the gain of the attacker when he performs an attack. The other quality to evaluate countermeasure node is the ROI, which is the return of investment. This metric extracts from annual expenses of security damages from attacks, the cost of security investment, and risk mitigation (RM) which shows the effectiveness of countermeasure and its value is from 0 to 1. Using both ROI and ROA simultaneously helps to better evaluate the effect of used countermeasures. Table 10 presents the summary of the whole methods to analyze the attack trees.

3.6.8 Attack Tree/Attack-Defense Tree Analysis

In [28] the authors formalize the bottom-up analysis of Attack-Defense tree using a mathematical approach. Assume we have a set of parent nodes (T) in the ADTree and the set L involves all the leaves of ADTree. For an attribute like $attribute_1$ in the $node_1 \in$ T, the α is a recursive function which assigns the attribute of the node to a value from set (V). The set V can be defined based on $attribute_1$, and the operation of α. The β is a function which assigns the attribute of a node from set L a value from set V. The children of $node_1$ makes the set C.

$$\alpha_{(node_1)} = \begin{cases} \beta_{(node_1)} & , node_1 \in \mathrm{L} \\ I_{\left(\alpha_{(node_\mathrm{i})},\dots,\alpha_{(node_\mathrm{k})}\right)}, & node_1 \in \mathrm{T}, node_\mathrm{i} \in \mathrm{C} \end{cases}$$

$I_{\left(\alpha_{(node_2)},\dots,\alpha_{(node_\mathrm{k})}\right)}$ is a function which defines the computational arithmetic based on the refinement of nodes. In this thesis we call this function ADTree creator function. This function should formulate all different refinements:

$$I_{\left(\alpha_{(node_2)},\dots,\alpha_{(node_\mathrm{k})}\right)} = \begin{cases} I_{(\vee^s)} & \text{, Disjunction between argumants , } s \in S \\ I_{(\wedge^s)} & \text{, Conjunction between argumants , } s \in S \\ I_{(c^s)} & \text{, 2 sibelling nodes from two different players} \end{cases}$$

Example

Consider the below ADTree on Fig 7. We are going to calculate the probability of successful attack in the root. In this case, the α function is the function which assigns a value from range [0,1] to a given node. The β function is derived based on historical data as following: $\beta_{(a)}$=0.2, $\beta_{(b)} = 0.7, \beta_{(d)} = 0.9$ *and* $\beta_{(e)}$=0.1. Also $I_{\left(\alpha_{(node_2)},\dots,\alpha_{(node_\mathrm{k})}\right)}$ is defined with:

$$I_{(\vee^s)} = \sum_{k=1}^{n} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,..,n\} \\ |I|=k}} \alpha_{(T_I)}$$

$$I_{(\wedge^s)} = \prod_{\forall\, i \in C} \alpha_i$$

$$I_{(c^s)} = I_{(x,y)} = \alpha_x * (1 - \alpha_y)$$

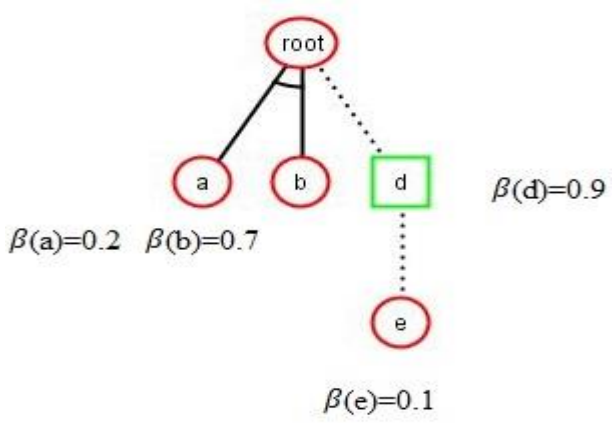


Fig 7. sample ADTree to perform bottom-up analysis

To evaluate the I(root), we need to follow the corresponding ADTrem of this ADTree which is: $c^{\boldsymbol{p}}$ ($\wedge^{\boldsymbol{p}}$ (a,b),$c^{\boldsymbol{o}}$ (d,e))

$I_{(I_{(a,b)},I_{(d,e)})}=I_{(a,b)} * (1 - I_{(d,e)})$

$I_{(a,b)}$=0.7*0.2=0.14

$I_{(c,d)} = I_{(a,b)} * (1 - I_{(d,e)})$=0.14*(1-0.9*(1-0.1))=0.0266

3.7 Summary

In this chapter, we have reviewed the different extensions of attack tree. The multiset semantic, and Boolean terms explain how an attack tree could be expressed with mathematical notation. There are also two main approaches to evaluate the severity of the risk. These two approaches are bottom-up, and Boolean function. Also, we defined the minimum acceptable level of security when the costs of attacks for an attacker is more than the benefits of it.

Table 11-The summary of some methods to analyze the attack trees.

Method	Approach		Metrics
	Bottom-up	Boolean function	
Multi-parameter		✓	P,π, $\boldsymbol{\pi}_-$,q , $\boldsymbol{q}_-$
Parallel Model		✓	P,π, $\boldsymbol{\pi}_-$,q , $\boldsymbol{q}_-$
Serial Model		✓	P,π, $\boldsymbol{\pi}_-$,q , $\boldsymbol{q}_-$
Fully Adaptive		✓	P,π, $\boldsymbol{\pi}_-$,q , $\boldsymbol{q}_-$
Attack Tree	✓		the probability of success
Attack-Defense tree	✓		the probability of success
Vulnerability Tree	✓		Attack Cost, Probability, Risk value
Protection Tree	✓		Risk, Impact, Cost, Probability
Defense Tree	✓		ROI,ROA

Part II: Contribution

4 Alignment of Attack-Defense tree to ISSRM

In this chapter, we answer

Q2.1.What attack tree constructs could be aligned to the ISSRM domain model?

As it is discussed before, the ISSRM by itself is only a class diagram in UML. It is also, important to improve the modeling languages in order to be able to show the concepts of ISSRM in design level. In this chapter, we are going to make an alignment from Attack-Defense tree to ISSRM.

4.1 ADTree Alignment to ISSRM

Asset and vulnerabilities: The asset nodes are extracted from the text description of the attacks, and the label analysis of attack tree. Sometimes vulnerabilities are not explicitly mentioned in the text, but the attack description reveals it. For instance, consider the label of an attack node is "intercepting the wireless signal". In this case, we can consider the wireless signal, and the wireless modem as assets, and the interceptable signal as a vulnerability.

Risk treatment concepts: We add **security requirements** using the ***countermeasure nodes*** on ADTree. These security requirements are written in "shall be" format, the rest of countermeasure nodes which are not in "shall be" format are considered as **control**. The cost attribute in the countermeasure node describes the expenses of implementation of control or security requirement. Also, the ***cut line*** acts as the **mitigate** association, when the countermeasure node is a child of an attack node.

Fig 8 shows the aligned ADTree (A-ADTree) which is capable of representing the main concepts of ISSRM. The combination of motive, capabilities, opportunities, and means make the threat agent. Since every attack node has a threat agent inside it explicitly, we connect the threat agent classes to the attack node using aggregation association. Additionally, the attack node exploits the vulnerabilities, and threaten the assets. Table 16. also, shows the alignment of new ADTree to the ISSRM concepts.

4.2 Attribute of Nodes

As the summary of the previous chapter showed, the probability of a successful attack is always considered as an attribute of nodes. This probability (P) can get extracted from the previous history of attacks on the system. Also, (Cost) of attack is important, because we are considering the analysis based on an economic framework. Cost can be evaluated from features of the attacker (Motive, Capabilities, Opportunities, Means), and the countermeasure node. Moreover, by applying the countermeasure nodes on an attack node the probability of success will decrease and the cost of attack will increase. In case the attacker successfully performs the attack he gains some benefit (Gain). We consider the value of attacked asset as a maximum gain of the attacker. Additionally, the countermeasure nodes have cost attribute, which shows the cost of implementation of it. This value cannot be more than the value of targeted asset. The following table shows the attribute of new AD Tree node. Table 12. Presents the equivalent concrete syntax in ISSRM, and A-ADTree.

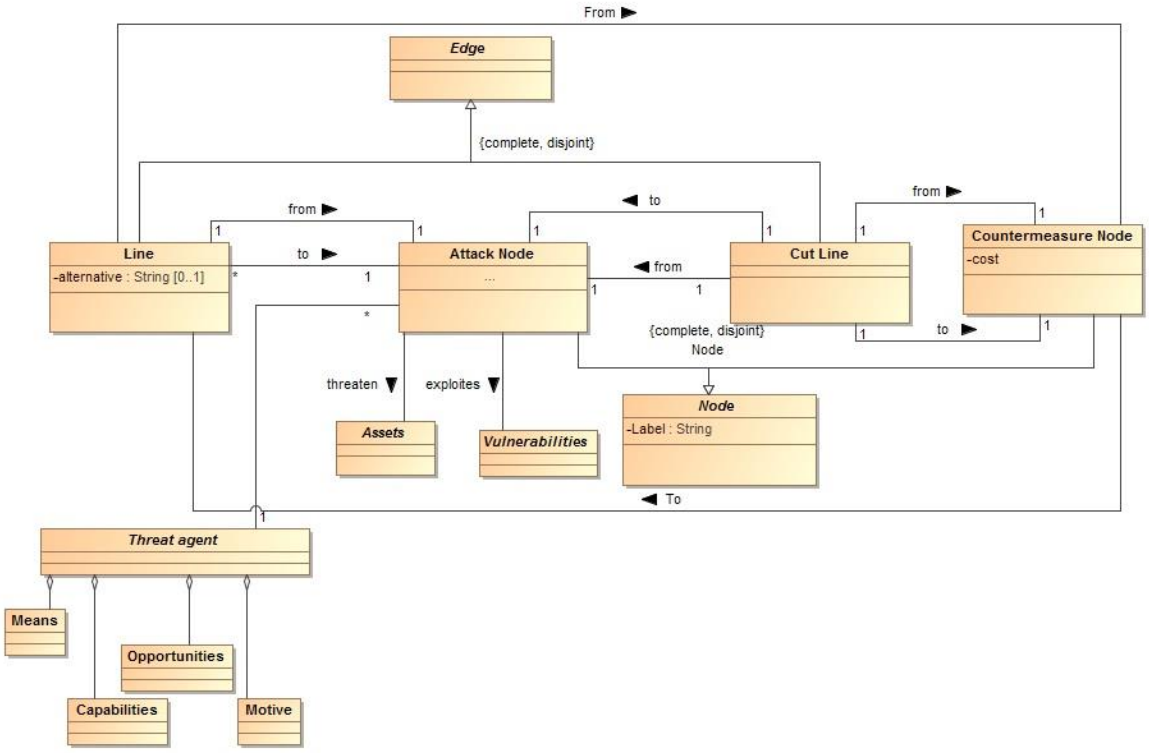


Fig 8. New A-ADTree abstract syntax

Table 12. alignment of A-ADTree to ISSRM

ISSRM	New ADTree
Security requirement Controls	Countermeasure node Line
Vulnerability	Vulnerability
Asset	Asset
Threat agent	Aggregation of motive, capabilities, opportunities, and means
Attack method	Aggregation of attack node, motive, capabilities, opportunities, and means, and line
Risk	Aggregation of vulnerability, attack node, motive, capabilities, opportunities, and means, and line
mitigate	Cut line

4.3 Summary

In this chapter, we tried to first align the context which covers similar concepts in ADTree, and ISSRM to each other. We found out, some of the concepts in the ISSRM are not covered in the ADTree. Therefore, we added these concepts to the ADTree. The final, ADTree component covers the concepts of the threat agent, asset, vulnerability, attack method, risks, and

risk treatment. We call this new ADTree as Aligned ADTree (A-ADTree). Table 13. explains different parameters in each node of A-ADTree, but we describe it more in detail in next chapter.

Table 13- Attribute of A-ADTree and the approach to evaluate

Node	Attribute	Evaluation
Attack Node	Outcome	Fully adaptive approach
	Probability of successful Attack (P)	Derived from historical data and OCTAVE
	Gain of successful Attack (Gain)	Asset Value
	Cost of attack	Derived from threat agent score of attacker, and countermeasures
Countermeasure Node	Cost of implementation ($\boldsymbol{C_c}$)	$\boldsymbol{C_c}$ < asset value
	Probability of success	Derived from historical data

5 Risk Severity vs Countermeasure Effectiveness Evaluation

Each node in the A-ADTree may have several metrics. In this thesis, we considered probability of success, and cost of the node. The added concepts such as vulnerability, and asset helps to measure these metric more precisely. In this chapter, we answer the below question.

Q2.2 What is the evaluation of metrics in both attack, and defense nodes?

5.1 Evaluation of probability using OCTAVE method

It is not possible to evaluate the probability of an atomic attack based log files or monitored measures in an information system. Vestiges of a given atomic attack may be found on several different measures (network log files, network traffic, firewall logs, OS services), and some of them may not get monitored in the information system. Additionally, it could be hard to evaluate a probability of a successful attack based only based on monitored measures, since some of the atomic attacks could be very abstract. On the other hand, we can evaluate the probabilities subjectively. In this approach, a person tries to estimate the probability of an event based on a set of evidence. Clearly, this approach is not precise, and a person may evaluate same events in the same situation differently, but we can increase the chance of correctness, by considering relevant measures. One of this most important information could be the attribute of the **threat agent** (capabilities, motive, opportunities, and means). Also, the person in the position of **network security**, and **a person familiar with the industry sectors** can evaluate this probability.

The OCTAVE [37] method is a combination of classical, and subjective evaluation of probabilities. This approach consists of three main steps:

1) Information gathering about threat agent, target assets, and the other factors which may affect the probability of occurrence of an attack. Also, there is a need to historical data about previous attacks, and periodical duration of each attack.
2) Define the criteria for evaluation of probabilities. In this thesis, we have defined the below criteria.

Table 14- Qualitative value of attacks

Very High	More than three times in a year
High	Two or three times in a year
Medium	Once in a year
Low	Three or four times in the last 5 years
Very Low	one or two times in the last 5 years

3) Assign the atomic attacks a qualitative value from the second step based on historical data of attacks, threat agent attributes, and expertise of selected people.

The previous three steps provide a qualitative evaluation of probabilities, but regarding this thesis, we need to a quantitative analysis of probabilities too. The below table provides the respective quantitative value for each criterion.

Table 15. Quantitative value of attacks

Qualitative value	Definition	Quantitative value
Very High	More than 16 times in last 5 years	90%
High	6 - 15 times in last 5 years	70%
Medium	Five times in last 5 years	50%
Low	Three or four times in the last 5 years	20%
Very Low	Zero, one or two times in the last 5 years	10%

5.2 Historical data format

We mention in the OCTAVE we need historical data, in this section, we introduce the format of this historical data. These data should be kept in three tables in a database. Fig 9 shows the table of data in database.

Asset Table: This table keeps the information of the targeted assets. Field times is the average number of times which this asset get attacked during the last 5 years. After each successful or not successful attack, this field should increment by one.

Table 16. Asset Historical Data

field	Description
Name	Name of asset
description	Description of asset
Value	Value of asset in a certain currency
Times	Average number of attacked on the asset

Threat Agent Table: We can consider each threat agent as a vector with four dimensions (Means, Motives, Capabilities, Opportunities). We evaluate motives, capabilities from 1 to 11. One is the lowest point and 11 is the highest point. The opportunity varies between 0 and 10. 0 means the threat agent has no opportunity to access the system (either remotely, or physically), and Infinite means the threat agent has no limitation for accessing the information system. Tables 1 to 4 shows the evaluation of each attribute.

Table 17. Evaluation of motivation attribute of threat agent-Adapted from [43]

	Motivation					
	Curiosity	Personal fame	Personal Gain	Revenge	National interest	Ideology
Score	1	3	5	7	9	11

Table 18. Evaluation of capabilities attribute of threat agent-Adapted from [43]

	Capabilities			
	Beginner	Undergraduate	Master	Specialist
Score	1	4	7	10

Table 19. Evaluation of opportunities attribute of threat agent

	Opportunities (Number of trial tries)			
	Zero	One	Finite	Infinite
Score	0	1	4	10

In [37], the author points to different kind of assets which can be considered here as means. In table 4 we considered 5 scores for each item. For instance, if the threat agent has three item of table 4, his score for means attribute will be 5*3=15.

Table 20. Evaluation of means attribute of threat agent-Adapted from [37]

	Means				
	Computer means	People as asset	Process as assets	Intangible assets	Stepstone assets
Score	5	5	5	5	5

Here computer means refers to, computer, laptops, networks and all the hardware infrastructure in the cyberspace. People in a different position and places like citizens, immigrants, sale people also can be considered as an asset. Additionally, some process like hiring employee, manufacturing are the means for the threat agent. In case that the threat agent has a reputation, stock price, intellectual property, he has some intangible tools. Finally, step stone assets are authentication data or remote network access.

Up to now, we have defined the attributes of a threat agent. Also, we have defined how can we evaluate each of these attributes. Now, to consider a general score for the threat agent, we can use the below formula:

Threat agent score= (Capability score+ Means Score+ Motivation score)*opportunity score/4

The multiplicity operation explains that in case that the threat agent does not have a trial chance for the attack, his skills, motivations, and means cannot effect his final score. On the other hand, if the threat agent has infinite trials then he would be more dangerous. The maximum score of threat agent is 115 and the minimum is 0.

Definition: Assume t is a threat agent vector, we define coefficient of threat agent using function g(t) as below.

Threat agent coefficient = *g(t)*= $Threat\ Agent\ Score/115$

Example: Assume for a certain attack the threat agent profile is estimated with:

(Means:25, Motives:9,Capabilities:6,Opportunities:4). Therefore, we have the following metrics:

*Threat Agent Score=(6+25+9)*4/(4*150)=0.26*

Attack Table: This table keeps the essential data regarding the attack. Here we introduce each field and we describe how to evaluate each one. Table 21. Shows the field of atomic attack table in database.

Impact table: This table describes what kind of effect it has on the targeted asset. The impact is not supported in the A-ADTree as a concept, but it helps the expert team to consider the impact when they are assigning the values to other concepts. The concept may have one of the values defined in ISSRM. Table 22. explains the fields of asset table in database.

Table 21. Attack Historical Data

Field	Description
Name	Name of attack method
Description	Explain how does this attack happen
AssetID	Foreign key of asset
Date	Date of attack
Success	Average Times attack successfully happened in last 5 years
Times	Average Times attack happened in last 5 years
CountermeasureID	Foreign key of countermeasure table
Cost of Damage (***Gain***)	Estimation of possible costs of attack damage by expert team
Cost of attack	Estimation of cost of attack by expert team

Table 22. Asset Historical Data

Field	Values
Impact	Loss of integrity Loss of availability Loss of confidentiality Loss of confidentiality, and integrity Loss of confidentiality, and availability Loss of integrity, and availability Loss of integrity, and availability, confidentiality
Description	Explanation of how impact of attack affects the system
Level	One, Two, Three

Vulnerability: This table contains the vulnerabilities of the asset. The A-ADTree and the ISSRM have covered the vulnerabilities of the asset. A vulnerability has a level value, which shows the severity of it. Table 23. explains each field of vulnerability table. Also, every vulnerability may lead to an impact. Therefore, there is a foreign key into the table.

Table 23. Field of vulnerability table in database

Field	Values
Name	Vulnerability name
Description	Explanation of how impact of attack affects the information system
Level	One, Two, Three
impactId	Foreign key of impact

Countermeasure Table: The countermeasure node could be any hardware or software which is applied on a certain attack method, in order to mitigate or avoid it. Every counter-measure may fail from its mission. Therefore, we can obtain the probability of success of it based on its failures.

Table 24. Countermeasure Historical Data

Field	Description
Name	Name of attack method
Description	Explain how does this attack happen
Cost	Cost of implementation of attack
AttackID	Foreign key of attack method which may apply
FailureTimes	Average time of failures in last 5 years Can be obtained from Success field of attack and Table 15

5.3 Evaluation of Probability of Attack

We mentioned that in OCTAVE we use both classical and subjective approach to quantify the value of probability. It would be very easy to use the table 15, **but it does not mention how the other parameters like threat agent profile, or target asset affect the value of probability.** To implement the effect of threat agent and the targeted asset, we used the **Bayes' theorem.**

Definition:

"Attack" is the event of an atomic attack performed successfully.

"Asset" is the event that a given asset is attacked.

In this case, the P(Atomic Attack |Asset) is the probability of atomic attack happens successfully, given the Asset was targeted.

$$\boldsymbol{P(Attack|Asset)} = \frac{\boldsymbol{P(Asset \cap Attack)}}{\boldsymbol{P(Asset)}}$$

Table 29. different statements needed for evaluating the P(atomic attack| asset)

Statement	Description
$P(Attack\|Asset)$	The probability of Attack, given the Asset, was targeted
$P(Asset \cap Attack)$	The probability of happening Atomic attack on Asset Can be obtained from join of Attack and Assets tables
$P(Asset)$	Can be obtained from Asset table, and Table 15

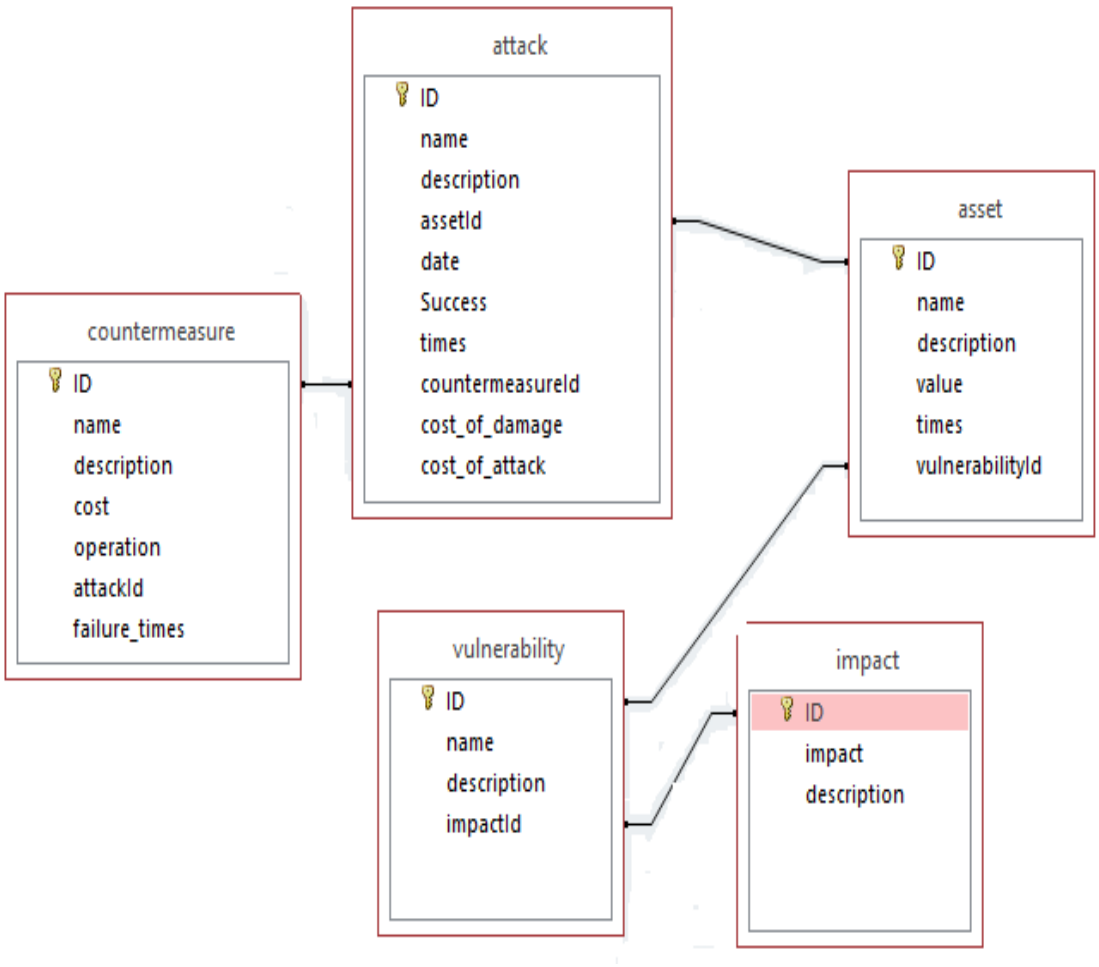


Fig 9. Relationship of historical data tables

Definition:

"Attack" is the event of an atomic attack performed successfully.

"Asset" is the event that a given asset is attacked.

"CTA" is the coefficient of a given threat agent.

We can apply the effect of threat agent on the probability of successful attack over an asset using, multiplication of threat agent coefficient on the probability of attack.

Probability of attack on asset consideration the threat agent profile= $P(Attack|Asset) * \text{CTA}$

5.4 Evaluation of probability of success in countermeasure node

$\boldsymbol{P_{countermeasure}}$ is the probability that the countermeasure node detect and resist against the attack and leads to failure of the attack. Fig 10 shows the classification of attacks.

$\boldsymbol{P_{countermeasure} = 1\text{-}Failure}$

Failure is the probability that the attack get detected by the countermeasure node, but the attacker performs the attack successfully. The Failure could be estimated using the counter-measure node and Table 16. The below image explains that we need to know about the success rate of atomic attack to evaluate the failure probability of countermeasures.

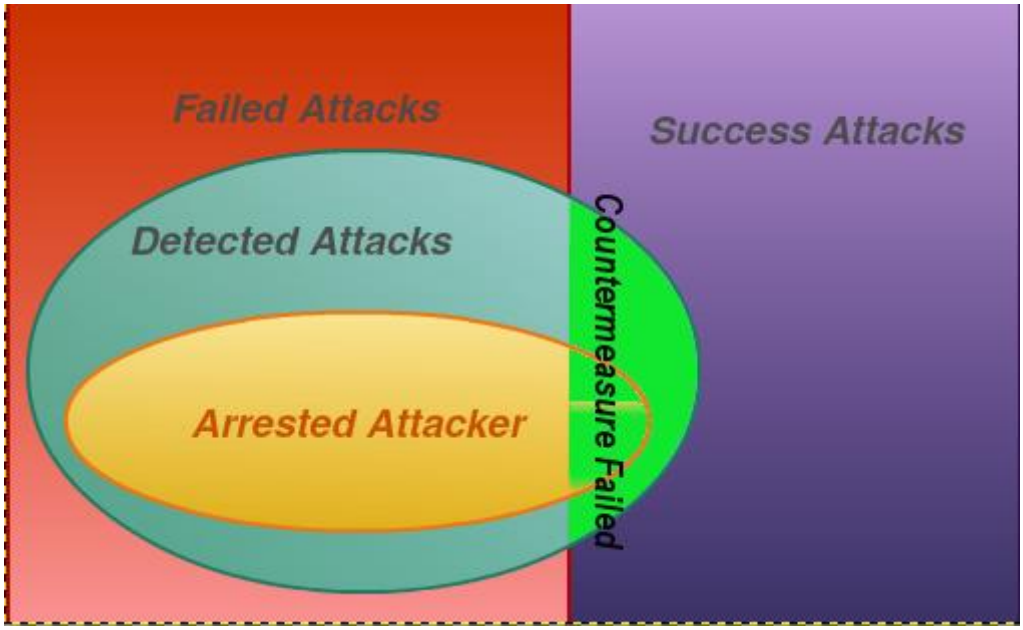


Fig 10. Classification of attacks

5.5 Example

Assume a threat agent with a profile such as { Means:25, Motives:9, Capabilities:6, Opportunities:4} attacked with DOS attack method to a router. The router was targeted 20 times in the last 5 years with different attacks. The DOS happened 12 times. DOS is one of the common attacks and has happened more than 30 times in the last 5 years on different assets in an information system. The company has set a firewall to mitigate this kind of attacks over the router since 5 years ago and it costs 8000€. The firewall has detected and stopped 10 out of 12, one attack was not detected, and one attack was detected but performed successfully DOS attacks on the router. Table 30. shows the evaluation probability of attack and countermeasure node.

Table 30. risk evaluation of example

Evaluation approach	Description
Atomic Attack	DOS
Asset	ROUTER
Threat Agent coefficient=0.26	(6+25+9)*4/4*150=0.26
P(ROUTER| DOS) =12/20=0.6	Fraction of DOS attack on all the attacks on router
P(DOS)=0.7	
P(ROUTER) =0.2	
Failure probability of firewall=1/12	Extracted from probability table

5.6 Summary

Figure 11 and table 31 explain the process of making an aligned Attack-Defense tree. There are few differences between this process and ISSRM and it comes from the fact that in A-ADTree we do not have some of ISSRM detailed classes like security objective, risk treatment, and control selection. The table describes, the whole process in separate steps with more details.

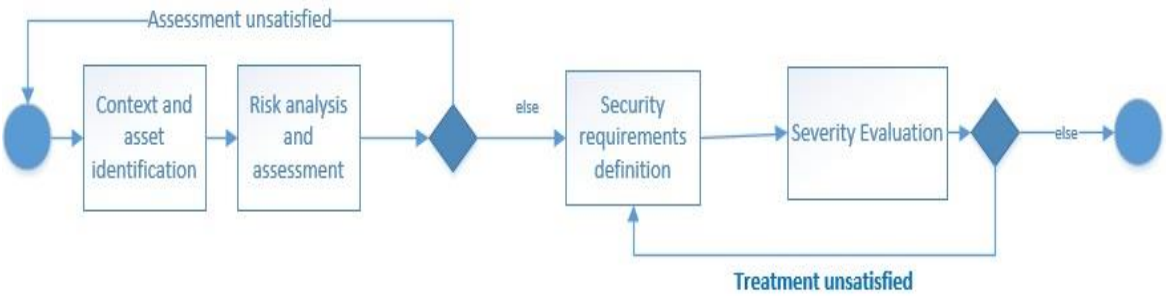


Fig 11- A-ADTree Process

Table 31-A-ADTree process

A-ADTree Process	Steps include
Context and asset identification	Build Asset table
Risk analysis	Build historical data
Security requirement and definition	Add countermeasure nodes on previous attack tree
Severity evaluation	Evaluate the parameters of each node

6 Bottom-Up Approach in Aligned Attack-Defense tree

In the previous chapter, we explained how an atomic node (either atomic attack, or a countermeasure) nodes get evaluated. In this chapter we discuss, how the effect of leaf nodes propagates up to the parent nodes, and the root. The approaches in this chapter is true both for attack, and countermeasure nodes.We answer

Q.2.3 What is the new measurement model in aligned Attack-Defense tree?

There could be three different approaches for threat agent to choose the nodes. Threat agent can choose based on cost, probability, or both (probability and cost)

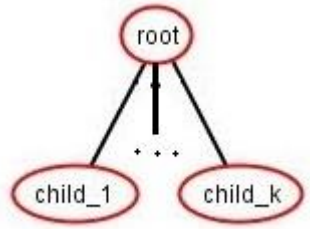
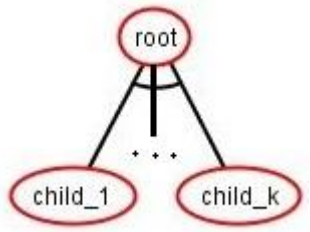
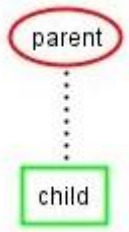
Assumption

Given that a node has children such as $\{child_1, child_2, \dots, child_n\}$. We point to the cost of the ***ith*** children with $cost_i$, and the probability of same children with pro_i

6.1 Cost Oriented Approach

In this approach, the threat agent tries to choose the option which makes the minimum cost. The cost of parent nodes with conjunction operation is equal to the sum of all children. But, when the parent has disjunction operation, the threat agent choose the child which has the minimum cost, regardless of its probability of success. Table 32. shows the cost evaluation of cost for attack nodes, but the same formula is used for countermeasure node.

Table 32. Minimum Cost for Attack Tree

Graphical presentation	*Selection metric*
	$\mathbf{cost_{root} = min\{cost_1, cost_2, .., cost_k, \}}$
	$\mathbf{cost_{root}} = \sum_{i=1}^{n} child_i.cost$
	Expert team should decide

Example: Figure 12 shows an attack tree which has two atomic attack nodes. The cost for the first child node is 300€ and the cost for the second child is 800€. Since the refinement in the parent node is Conjunctive (AND), the cost for the parent node is 300+800=1100. Also, there is an countermeasure in the root node which increases the costs of the root. The expert team can evaluate the effect of the countermeasure on the cost.

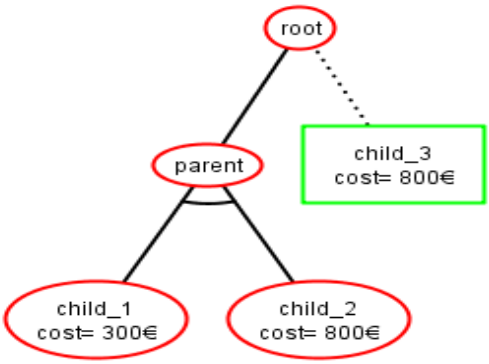


Fig 12. Example of attack-defense tree with given costs on atomic nodes

6.2 Probability oriented approach

In this approach, we choose the options which make the maximize probability of success. The probability of parent nodes with conjunction operation is equal to the multiplication of all probability children. Here we assume the atomic actions are independent. But, when the parent has disjunction operation, the threat agent choose the child which has the maximum chance, regardless of its cost. Table 33. shows the probability evaluation of cost for attack nodes, but the same formula is used for countermeasure node.

Example: Figure 13 shows an attack tree which has two atomic attack nodes. The probability of the first child node is 0.3 and the probability of the second child is 0.4. Since the refinement in the parent node is Conjunctive (AND), the probability of the parent node is 0.3*0.4=0.12. Also, there is a countermeasure in the root node which decreases the probability of the root. Regardless of the countermeasure the probability of the root node comes from its only attack child (0.12). Considering the countermeasure the probability of the root is equal to 0.2*0.12=0.024

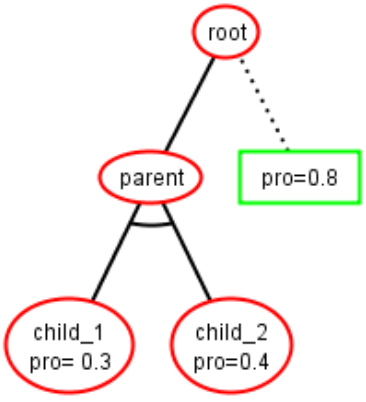
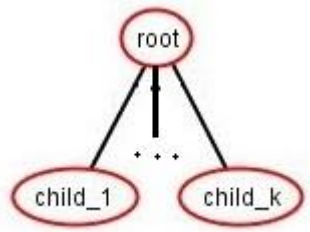
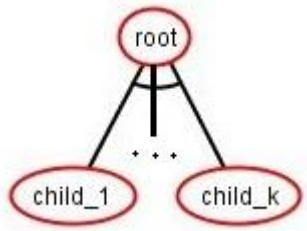
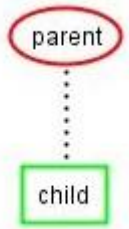


Fig 13. Example of attack-defense tree with given costs on atomic nodes

Table 33. Maximum Probability for Attack Tree

Graphical meaning	*Selection metric*
	$\mathbf{pro_{root} = Max\{pro_1, pro_2, .., pro_k,\}}$
	$\mathbf{pro_{root}} = \prod_{i=1}^{k} \boldsymbol{children_i.pro}$
	***Assume:*** probability of parent node regardless of the counter-measure is $\mathbf{original - pro_{parent}}$ $\mathbf{pro_{parent} = original - pro_{parent} * (1 - pro_{child})}$

6.3 Summary

In this chapter, we discussed how bottom-up approach happen in the aligned Attack-Defense tree. In the first step, the probability and cost of atomic nodes should be evaluated. This evaluation should happen both for the countermeasure, and attack nodes, and we have discussed it in the previous chapter. Then, there could be two different approaches for both attacks and defense trees. The approaches of attack tree are independent of approaches in countermeasure nodes. It means that, although the threat agent may choose its options based on probability, the defender team may choose options based on minimum cost.

Part III: Validation (proof of concept)

7 Major Information About the Implemented Tool

During this master thesis, we have developed a tool to shows the aligned ADTree. We used java, sqlite for storage. The source code of this tool exists in this repository.The basic of this project has been developed by authors of [25].We have continued this open source project and added some features to it in order to be able to show an aligned ADTree.The basic code of the ADTool exists in this link.In this chapter, we answer

Q3.1 What is design for implementation of aligned Attack-Defense tree?

7.1 Major Usecases of the Tool

Table 34. shows the main functional requirements of A-ADTree. The main usecases in the implemented aligned ADTree is about importing data related to the atomic attack, and countermeasure, and then create an ADTree based on them. The basic data to make an ADTree are the impact of an attack, vulnerability of assets, assets, atomic attack and atomic countermeasures. Once the Aligned ADTree is made, the user is able to assign each atomic node an atomic action. Atomic actions (atomic attack, or atomic countermeasure) are already inserted in the database. At the end, the user can obtain the level of the risk in each attack node, by seeing the properties of each node.

Table 34- Functional Requirements of A-ADTree

ID	*Requirement*
1	The A-ADTree shall be able to keep impact data with name, level, and description of impact
2	The A-ADTree shall be able to keep vulnerability data with name, level, and description of vulnerability
3	The A-ADTree shall be able to keep asset data with name, vulnerability, description, and value of the asset.
4	The A-ADTree shall be able to keep atomic attacks data with name, description, the cost of attack, the cost of treatment, and the probability of success of attack method of it.
5	The A-ADTree shall be able to keep atomic countermeasure data with name, description, cost of implementation, and probability of success
6	The A-ADTree shall be able to show the profile of threat agent based on his capability, motivation, means, and opportunities.
7	The A-ADTree shall be able to suggest values to assign to atomic nodes
8	The A-ADTree shall be able to connect the visual nodes as a tree to each other
9	The A-ADTree shall be able to show the probability of success of attack nodes
10	The A-ADTree shall be able to show the cost of attack node
11	The A-ADTree shall be able to show the effect of threat agent profile on attack nodes
12	The A-ADTree shall be able to show the effect of countermeasure nodes on attack nodes.
13	The A-ADTree shall be able to suggest options to select the children on disjunctive nodes

7.2 List of Requirements

Here we briefly enumerate the list of requirements of the aligned ADTree(A-ADTree). Each requirement is linked to the respected usecases in Appendix II.

7.3 Architecture of Tool

There are three main components in the A-ADTree.

1) The database which is responsible for keeping the data respected to the impact, vulnerability, asset, atomic attacks, and atomic countermeasures.
2) Computational Component which is responsible for evaluating the risk, the severity of the risk, the probability of success of nodes. It also contains information about the structure of ADTree.
3) A graphical component which is responsible for presenting the attack nodes, and countermeasure nodes.

Figure 14 explains how these three components are related to each other. Every time the user make a change on the graphical component, these changes are applied to the database, and structure of the tree.

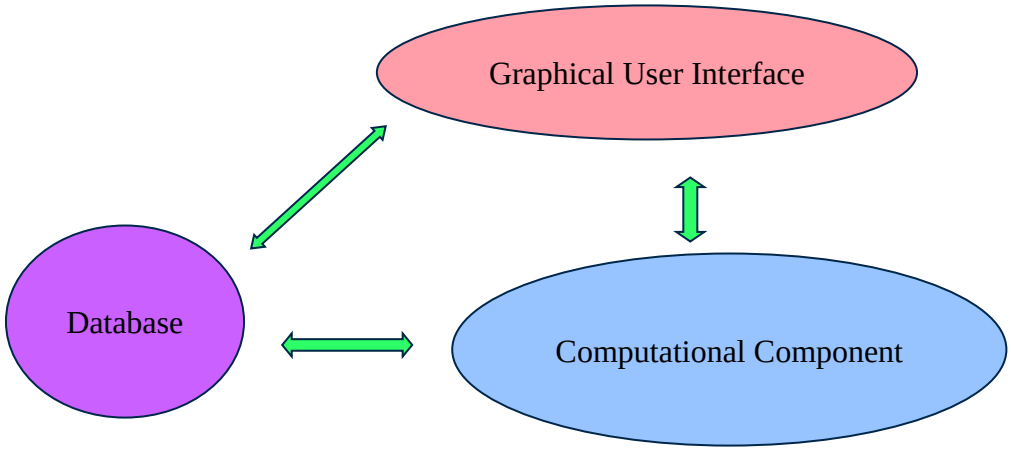


Fig 14. Abstract architecture of the A-ADTree

7.4 Limitation of the Tool

The A-ADTree does not cover the evaluation of probability based on historical data which is covered in part 5.3. Here, we made an assumption that the users (expert team) evaluate the probability of attacks and cost of the atomic actions. But, the rest of the nodes in the A-ADTree obtains their values based on cost oriented, or probability oriented approaches which is introduced in chapter 6.

7.5 Summary

In this chapter, we tried to briefly explain the main requirements, usecases, architecture, and limitation the of the implemented A-ADTree. The developed tool was based on an open source tool provided in [25]. Also, the functional requirements, and the behavior of the A-ADTool are explained well in the appendix with usecase diagrams.

8 Examples of Usage of Aligned ADTree

This chapter answer to the

Q3.2. What are the examples of risk mitigation using aligned Attack-Defense tree?

In [44] the authors provide two examples about evaluating the risk severity using the capability of a threat agent. They showed that the higher capabilities the threat agent has, the higher probability exist for the risk to happen. In this chapter, we are going to illustrate the same examples using A-ADTree, and then we show how countermeasures can mitigate the risks.Before we continue to the next step, we need to define the equivalent terminology in [44] and the concepts in A-ADTree. Table 35. shows how terms are equivalent.

Table 35-Equivalent term in [44] and this thesis

	Concept in [44]	Concept in Aligned ADTree
1	Capability	Atomic Attack
2	Score	Cost of action
3	Capability Likelihood	Probability of success in attack method
4	Risk Severity	Sum of cost of treatment

8.1 Risks on Video Conference System

Video conference (VC) communication is a common tool in the daily business meeting and online lectures. The VC provide all the users in the same session the ability to read, write, and execute the shared documents. Also, users in a VC session can delegate each other to remotely control each other device, and camera.

A malicious threat agent may intercept the channel between the users and eavesdrop over the channel. In this case, he is able to read or write some shared documents. The below table shows the likelihood, and score, for each capability of threat agent [44].

The examples in [44] only provide risks and attack methods, but here we add some countermeasures to each of the risks or the attack methods. The crypto-functionalities can be used to protect the integrity and the confidentiality of the message. Also, with replacing the media with a new media which is not intercepted able can prevent man in the middle attacks. Table 36. explains the likelihood, and score of each attack method in [44].

Table 36. Correlation between score and likelihood in [44]

ID	Capability	Score	Likelihood
C1	Discover an active station in VC	4	0.8
C2	Eavesdrop the communications in the channel	4	0.8
C3	Modify sent messages	3	0.5
C4	Change installed software using an installed malware	2	0.2
C5	Access to power in or off of the station	2	0.2
C6	Modify the hardware using physical access	1	0.01

Table 37. Countermeasures against the attack methods

ID	Countermeasure	Cost Of implementation	Probability of failure
Counter1	Authenticate all join requests	2000	0.02
Counter2	Replace channel with a new un-interceptable one	50000	0.03
Counter3	Use crypto functionality in sent messages	1000	0.05
Counter4	Use anti-spyware system on the VC stations	5000	0.01
Counter5	Authenticate people before physical access	20000	0.01

Based on the above capabilities of the threat agent, table 38 describes following threats. Also, table 37 include the countermeasure which may apply on the attack methods.

Table 38. Correlation between threat, and capabilities [44]

Threat ID	Threat Description	Risk Severity	Capabilities
T1	Join a meeting without authentication	11	C1, or C4, or C6
T2	Control the hardware like camera without authentication	13	C3,or C4,or C6
T3	Intercept messages in channel	11	C2,C4
T4	Intercept messages in channel, and modify them	11	C3,C4,C6
T5	Interrupt the VC station using DDOS attack	6	C3,or C4,or C5,or C6
T6	Update firmware likes (ROM) without authentication	10	C4, or C6

8.1.1 Structure of Attack Tree for VC Risk

Here we use the data in [44] to make a tree in aligned ADTree. Figure 15 shows the skeleton of ***all risks in a video conference station,*** the risk can happen based on each of threats like T1, T2, or T3. The refinement of each threat node is based on table 36. We configure the atomic node, C1, as figure 16 shows. Then the properties of the root node in the figure 18. We can find the effect of threat agent profile, using the right-hand side border. In the very first step, we have chosen the "maximum probability" as a metric for choosing the child in the parent with disjunctive refinements.

In VC examples, the T1, and T2 nodes have the maximum probability, therefore, they propagates their probability to the parent which is the root node.

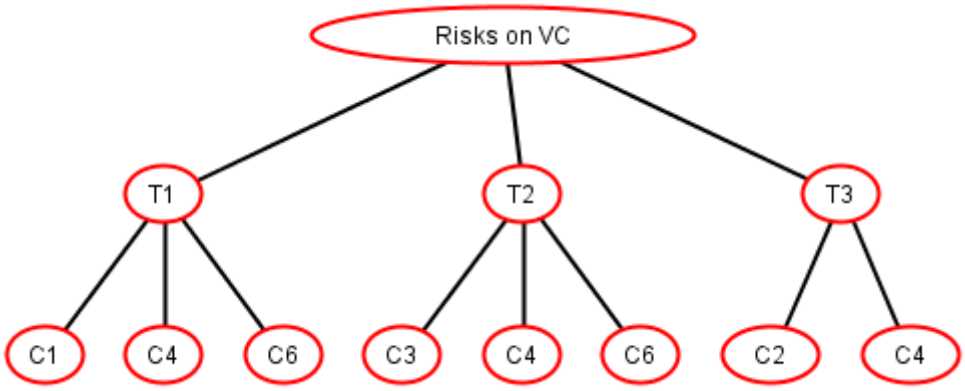


Fig 15. Structure of attack tree in VC example

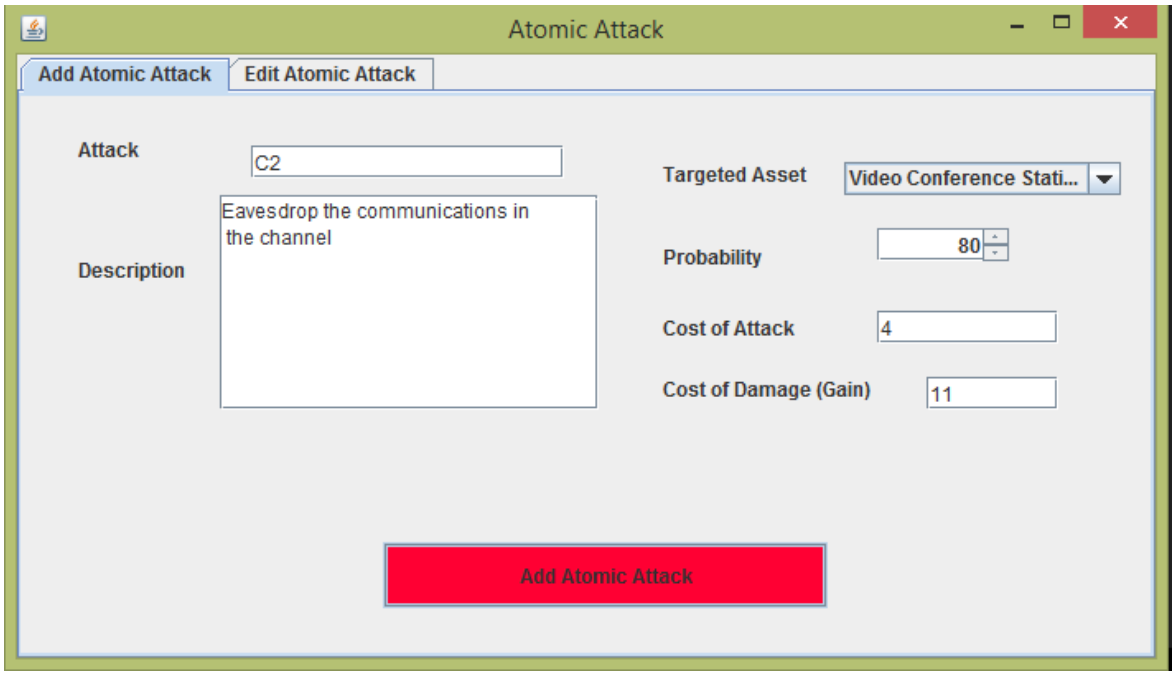


Fig 16. Configuration of atomic nodes

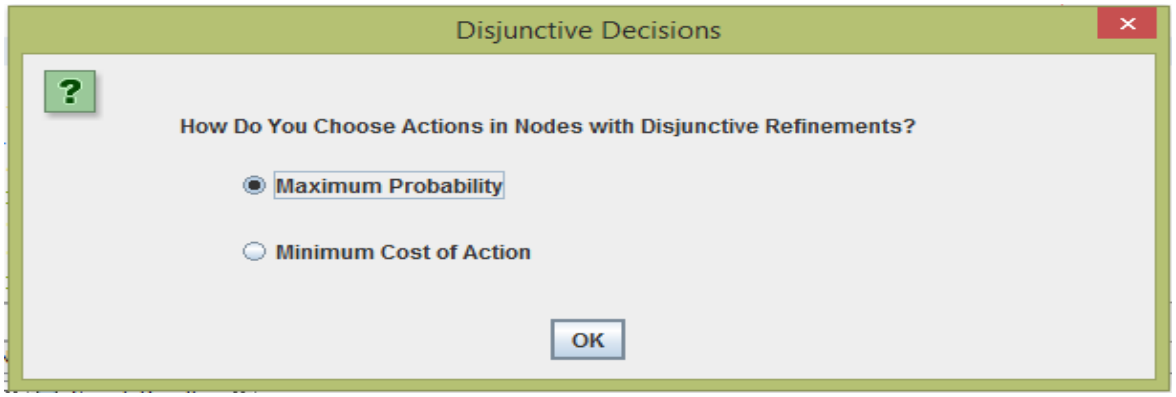


Fig 17. Selection criteria in nodes with disjunctive refinement

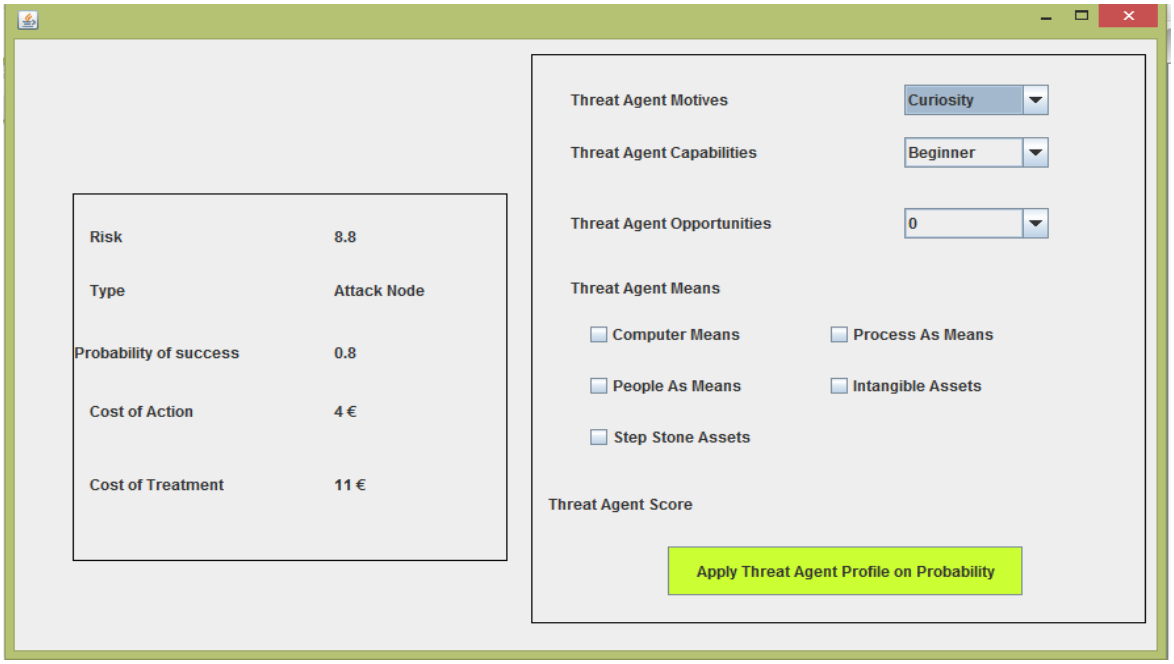


Fig 18. Evaluation of risk regardless of threat agent profile

8.1.2 Effect of Threat Agent Profile on Risk

A threat agent with the profile like motive: personal fame, capabilities: undergraduate, opportunities: finite, and means: computer means, process, people, and intangible, obtains a score close to 0.25.We apply the threat agent profile on the risk evaluation. The result shows a significant risk reduction from 8.8 to 2.06. Such result is because the threat agent evaluation is about 0.25 of maximum. Figure 19 shows the properties of attack node after applying the threat agent profile on it.

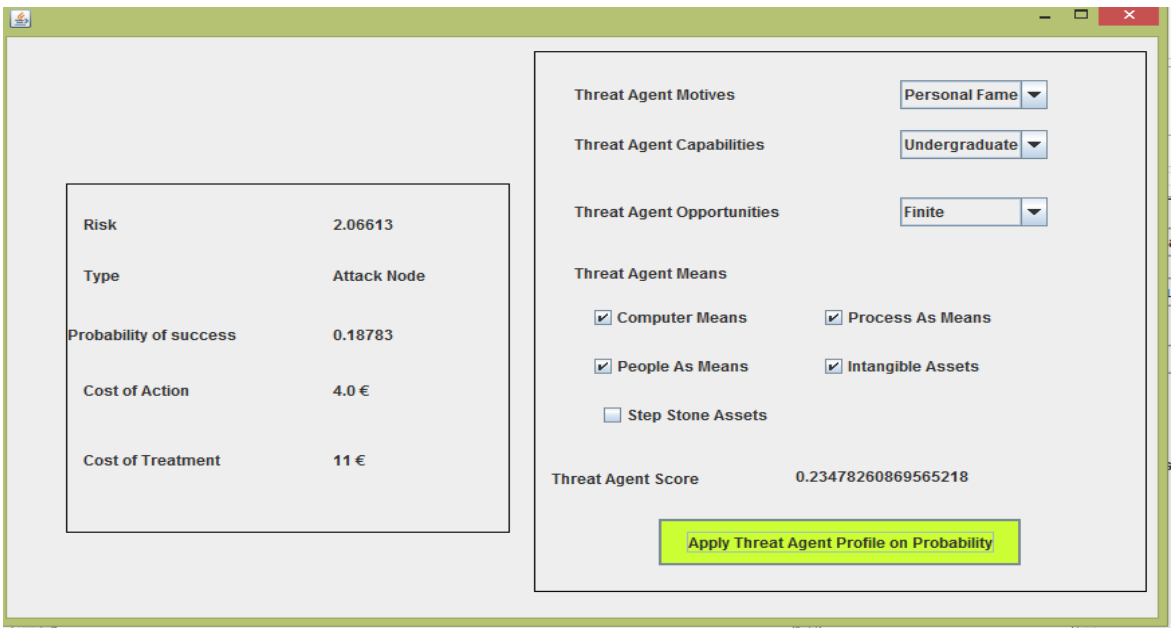


Fig 19. Evaluation of risk in presence of threat agent profile

8.1.3 Effect of Countermeasure Trees

Then we can apply the countermeasures to see the effect of countermeasures on the risk. The labels on countermeasures are based on Table 36. Figure 20 shows the structure of A-ADTree after applying the countermeasures. It shows the risk reduces from 8.8 to 2.64 after applying the countermeasures.

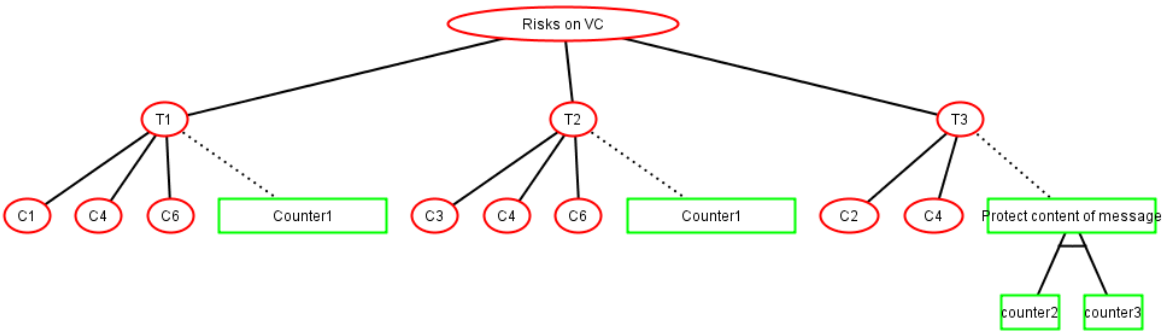


Fig 20. Structure of A-ADTree in VC example

Risk	2.64
Type	Attack Node
Probability of success	0.24
Cost of Action	4 €
Cost of Treatment	11 €

Fig 21. The effect of countermeasure on properties of root

8.2 Connected Vehicles

A connected vehicle contains some sensors. These sensors help the driver to control some operations in driving. Also, the connected vehicle may connect with personal devices, and receive the orders from the laptop, or cellphone.

Clearly, a threat agent can physically, or remotely find access to the sensors and manipulate the operations on the vehicle. For instance, the threat agent may affect the braking system with updating the Engine Control Unit (ECU). Such disruption leads to an accident ,when the vehicle is moving. Table 39 shows, the atomic actions of a threat agent in [44]. Also, table 40 shows how different capabilities of threat agent leads to a risk, based on [44]. Additionally, table 41 shows the countermeasures of the vehicle.

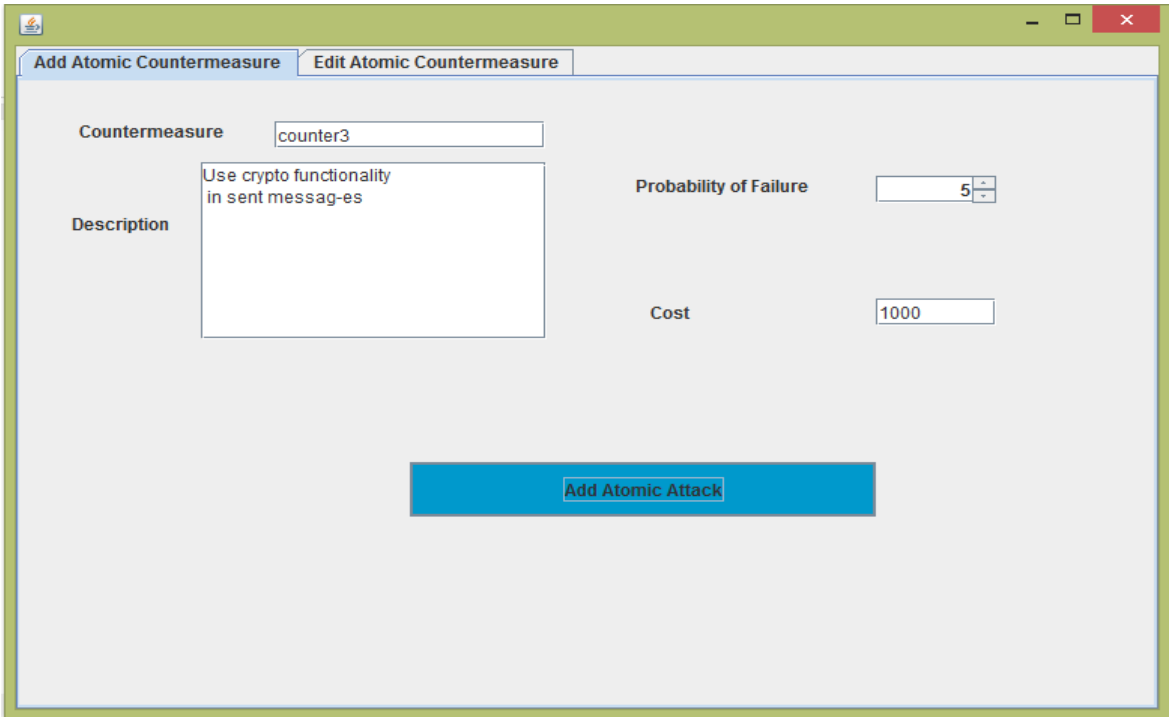


Fig 21.Adding countermeasures to atomic actions

Table 39. Atomic actions for threat agent

ID	Capability	Score	Likelihood
C1	Threat Agent has physical access to ECU	3	0.5
C2	Threat Agent can remotely inject message to CAN bus	2	0.2

Table 40- Correlation between threat, and capabilities [44] for connected vehicle

Threat ID	Threat Description	Risk Severity	Capabilities
T1	Remotely update ECU	12	C2
T2	Disruption of the braking system of the vehicle	10	C1, or C2

Table 41. Countermeasure solutions for connected vehicle

ID	Countermeasure	Cost Of implementation	Probability of failure
Counter1	Authenticate all join requests	4000	0.02
Counter2	Replace channel with a new un-intercept able one	30000	0.03

8.2.1 Structure of Attack Tree for Connected Vehicle

The below image shows how threat agent can use his atomic actions in table 37 to materialize the risks in table 38. For instance, to manipulate the braking system, the threat agent may modify the ECU physically or apply a man in the middle attack method on the vehicle.

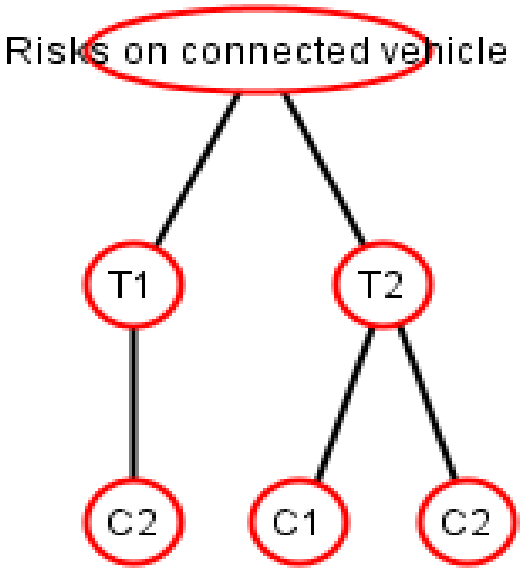


Fig 23. Structure of attack tree in connected vehicle example

Figure 24 shows the evaluation of the risk regardless of threat agent profile, and possible countermeasures.

Risk	6.0
Type	Attack Node
Probability of success	0.5
Cost of Action	3 €
Cost of Treatment	12 €

Fig 24. Evaluation of risk over the connected vehicle case regardless of threat agent

8.2.2 Effect of Threat Agent Profile on Risks

We apply the same profile in the previous example to this one too. Therefore, the threat agent with the profile like {motive: personal fame, capabilities: undergraduate, opportunities: finite, and means: computer means, process, people, and intangible} , obtains a score close to 0.25. The result shows that the profile of threat agent reduces the risk.

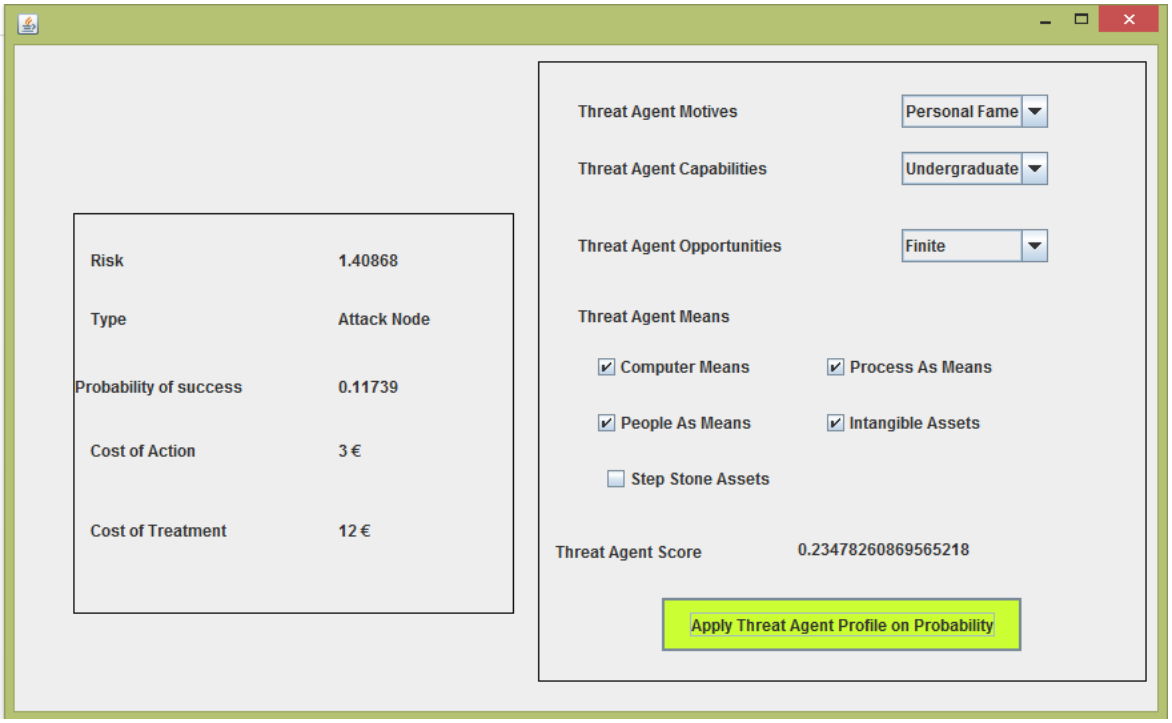


Fig 25. Evaluation of risk in presence of threat agent profile

8.2.3 Effect of Countermeasure Tree

We can apply the stated countermeasures to each nodes of attack tree in order to mitigate the risks.

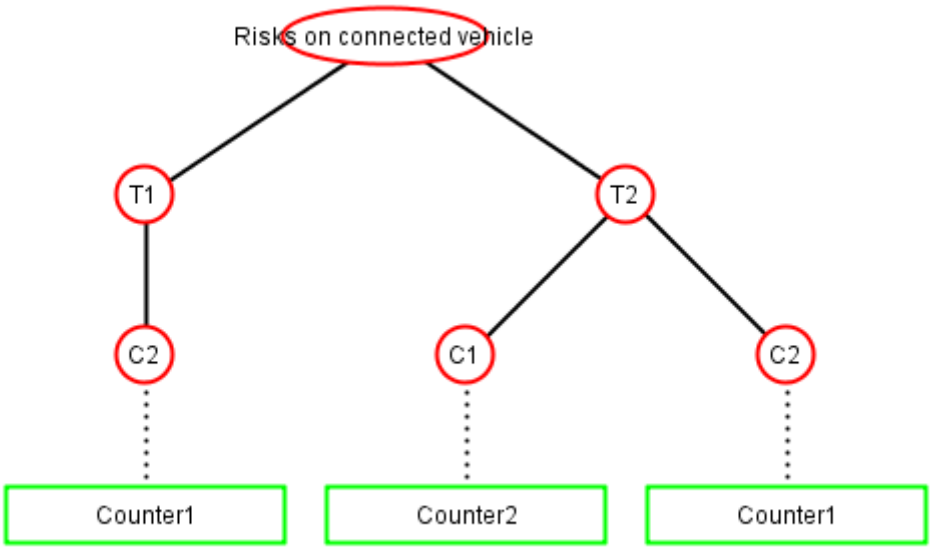


Fig 26. The countermeasures are applied on attack tree.

Figure 27 shows the effect of countermeasure trees to the evaluation of the risk. It shows that, the risk is reduced from 0.6 to 0.36 after applying the countermeasures.

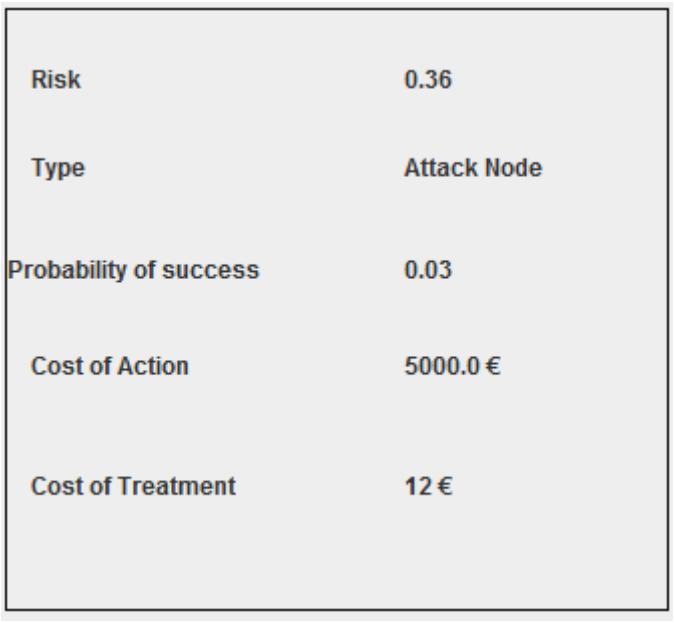


Fig 27. Risk evaluation after adding the countermeasures

8.3 Summary

In [44] the authors tried to measure the risk based on the threat agent profile. A threat agent profile in [44] includes the means, opportunities, and capabilities. Since, the threat agent profile in this thesis, and [44] has different parameters, the measurement of the threat agent is different. In this chapter, we used the examples in the [44] to show how the aligned AD-Tree works. In [44] the author shows the effect of threat agent on risk evaluation. Based on it, the risk is higher when the threat agent has advanced skills. Here, we also obtained the same results with same data. We showed, although the evaluation of threat agent differs, lack of skills of threat agent reduces the final value of risk.

9. Conclusion

In this thesis includes three main parts. The first part contains literature review about Information System Security Risk Management (ISSRM)[1,2,44], and some of the common modeling languages such as Secure BPMN [4,5,7], Secure TOROPOS [9,10], Misuse case [13], and Mal-Activity diagrams [18], and their alignment to ISSRM. Additionally, we have reviewed the family of Attack Trees, specifically, the Attack-Defense Tree which covers two main concepts of ISSRM by its nature. Also, we have reviewed, some of the common measurement approaches of risk using attack trees.

The second part belongs to the contributions. We first tried to obtain an alignment from Attack-Defense Tree to ISSRM. We called the obtained tree Aligned Attack-Defense Tree (A-ADTree). The ADTree covers risk concepts, and risk treatment partially, but some of the element are missed. The alignment helped us to detect the missed concepts of ISSRM in ADTree. Then we have added the missed elements into the ADTree, and obtained the A-ADTree. Some concepts such as asset, vulnerability, and threat agent are added to ADTree. We used the added concepts to measure the risk concept. The risk measurement includes measuring the probability (likelihood), and cost of the risk. We have involved the asset, and vulnerability into the measuring the probability. Also, we can obtain the probability considering the profile of threat agent who performed the attack method. At the end we have introduced how the values propagates in a bottom-up approach in A-ADTree.

In the last part, we introduced the tool that we have implemented to show how the A-ADTree works. This tool is based on the implementation of ADTree in the [25]. We have introduced the functional requirements, and the usecase of the A-ADTool. In the last chapter, we tried to use the data, and examples in [44]. In [44] the authors showed that the profie of the threat agent affects the risk.We obtained the same results in [44] using A-ADTree.

9.1 Assumption to requirements

In this thesis, we have assumed that the atomic nodes are independent. Therefore, the cost of the atomic nodes do not cover each other. Additionally, they do not effect the probability of each other. These kind of assumptions, facilitate the bottom-up approach of measuring the risk, but usually in the real world the used attacks are not independent from each other.

Also, in the implementation of the A-ADTree, we have assumed the probability of the atomic nodes are already given. Therefore, we did not involved in the evaluation of probability using the OCTAVE [37] method and historical data as it is proposed in the second part.

9.2 Conclusion

In this thesis, we have proposed an improved Attack-Defense Tree which can show the different concepts of ISSRM. Additionally, this model is able to measure the risk based on its elements (attack method, asset, vulnerability, and countermeasure). Also, the implemented tool is able to show the effect of threat agent on the measurement of the risk. We showed that, if the threat agent has the advanced profile, the probability of the risk will increase.

9.3 Answer to Research Questions

Here we briefly answer to the research questions.

Q1. How can we show the main elements of risks in information systems?

We have aligned Attack-Defense Tree to ISSRM, and detected that asset, vulnerability, and threat agents are not covered by ADTree. We added these three concepts to the attack nodes. As a result, the attack node show the attack method, and risk concepts. On the other hand, the countermeasure node covers the risk treatment concepts of ISSRM.

Q2.How can we align Attack-Defense tree to ISSRM?

Since, we have added the vulnerability, asset, and threat agent to the attack node, we can evaluate the probability of the risk based on these data. We used the OCTAVE to subjectively evaluate the probability of attack node, but later we used the historical data, and Bayes theorem to calculate the probability quantitatively.

Q3. How can we validate the aligned Attack-Defense tree?

We have extended the code on the [25], to add a database which gets its data from historical data. Also, we have added some features to show that every node has a cost and probability. The cost and probabilities make the properties of each node. The implemented A-ADTree is capable to show the effect of threat agent and countermeasure nodes on properties of attack nodes.

In [44], the authors provide two examples, and showed the effect of threat agents on the ISSRM. [44] showed that, the risk of an attack method has a direct relation with potentiality of threat agent. We used the same examples, and obtained the same results.

9.4 Future Work

In future, we obtain alignment from A-ADTree to the alignment languages, such as Secure BPMN, or Mal-Activity diagrams. These models also have been aligned to ISSRM. Therefore, we can find what elements in different modeling languages have similar semantic meaning based on ISSRM.

Additionally, it is important to **automatically** collect complete historical data. We may obtain this data from log files of the information assets. In this part, we can obtain this data using machine learning technique.

References

[1] Dubois, E., Heymans, P., Mayer, N., Matulevicius, R.: A systematic Approach to Define the Domain of Information System Security Risk Management, pp.289-306. Springer (2010).

[2] Mayer, N .:Model Based Management of Information System Security Risk. Phd thesis, University of Namur (2009)

[3] Mayer,N., Dubois, E., Matulevicius, R., Heymeans, P.: Toward a measurement framework for security risk management. In: Proceeding of the workshop on modeling security (MODSEC08) held as part of the MODELS 2008 (2008)

[4] Silver, B .: BPMN Method and Style: A Levels-based Methodology for BPMN Process Modeling and Improving using BPMN 2.0 Cody-Cassidy Press (2009)

[5] Dumas, M., La Rosa, M., Mendling, J., Reijers, H.: Fundamental of Bussiness Process Management. Springer (2013)

[6] Diallo, M.H., Romero-Mariona, J., Sim, S.E., Richardson, D.J.: A Comparative Evaluation of Three Approaches to Specifying Security Requirements. REFSQ'06, Luxembourg (2006)

[7] Altuhhova, O., Matulevicius, R., Ahmed, N.: An Extension of Business Process Model and Notification for Security Risk Management. International Journal of Information System Modeling and Design (IJISMD) 4(4), 93 – 113 (2013)

[8] Bresciani, P., Perini, A., Giorgini, P., Fausto, G., Mylopoulos, J.: TROPOS: an Agent-oriented Software Development Methodology. Journal of Autonomous Agents and Multi-Agent Systems 25, 203 – 236 (2004)

[9] Mouratidis, H.: A Security Oriented Approach in the Development of Multiagent Systems Applied to the Management of the Health and Social Care Needs of Older People In Eng-land. Ph.D. thesis, Department of Computer Science, University of Sheffield, UK (2004)

[10] Mouratidis, H., Giorgini, P.: Secure Tropos: A Security-oriented Extension of the Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering (IJSEKE) 17(2), 285–309 (2007)

[11] Mouratidis, H., Giorgini, P., Manson, G.: Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems. In: Proceedings of the 15th Conference On Advanced Information Systems Engineering (CAiSE'03). pp. 63–78. Springer-Verlag (2003)

[12] Matulevicius R. "Fundamentals of Secure System Modelling",(unpublished draft), 2016

[13] Sindre, G., Opdahl, A.L.: Eliciting Security Requirements with Misuse Cases. Requirements Engineering Journal 10(1), 34–44 (2005)

[14] Matulevicius R. "Fundamentals of Secure System Modelling", Chapter 9 (unpublished draft), 2016

[15] Schneier, B.: Attack Trees. Dr. Dobb's Journal (1999)

[16] Donald Knuth. The Art of Computer Programming: Fundamental Algorithms, Third Edition. Addison-Wesley, 1997. ISBN 0-201-89683-4 . Section 2.3: Trees, pp.308–423

[17] Sandip C. Patel, James H. Graham, and Patricia A. S. Ralston. Quantitatively assessing the vulnerability of critical information systems: A new method for evaluating security enhancements. International Journal of Information Management, 28(6):483–491, December 2008.

[18] Sindre, G.: Mal-Activity Diagrams for Capturing Attacks on Business Processes. In: Requirements Engineering: Foundation for Software Quality. pp. 355–366. Springer (2007)

[19] Börger, E., Cavarra, A., Riccobene, E.: An ASM Semantics for UML Activity Diagrams. In: Proceedings of the 8th AMAST 2000. pp. 293–308. Springer (2000)

[20] OMG: Unified modeling language: Superstructure, version 2.0 (2004)

[21] Kenneth S.Edge, George C. Dalton II, Richard A. Raines, Robert F.Mills: Using Attack and Protection Trees to Analyze Threats and Defenses to Homeland Security.

[22] Wand, Y., Weber, R.: On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. Journal of Information Systems 3, 217–237 (1993)

[23] Opdahl,A.L.,Henderson-Sellers,B.: A Unified Modelling Language without Referential Redundancy. Data and Knowledge Engineering (DKE), Special Issue on Quality in Conceptual Modelling pp. 277–300 (2005)

[24] Edge, K. S., Dalton, G. C., Raines, R. A., & Mills, R. F. (2006, 23-25 Oct. 2006). Using Attack and Protection Trees to Analyze Threats and Defenses to Homeland Security. Paper presented at the MILCOM 2006 - 2006 IEEE Military Communications conference.

[25] Kordy, B. Kordy, P. Mauw, S. Schweitzer, P: ADTool: Security Analysis with Attack-Defense Trees (Extended Version), white paper, Accessed on: 19 Aguste 2016 on: https://www.researchgate.net/publication/236955204_ADTool_Security_Analysis_with_Attack-Defense_Trees_Extended_Version

[26] Kordy B, Kordy P, Mauw S, Schweitzer P. ADTool: Security Analysis with Attack–Defense Trees. In: Joshi K, Siegle M, Stoelinga M, D'Argenio PR, editors. Quantitative Evaluation of Systems: 10th International Conference, QEST 2013, Buenos Aires, Argentina, August 27-30, 2013 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 173-6.

[27] Kordy B, Mauw S, Melissen M, Schweitzer P. Attack–Defense Trees and Two-Player Binary Zero-Sum Extensive Form Games Are Equivalent. In: Alpcan T, Buttyán L, Baras JS, editors. Decision and Game Theory for Security: First International Conference, GameSec 2010, Berlin, Germany, November 22-23, 2010 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 245-56.

[28] Kordy B, Mauw S, Radomirović S, Schweitzer P. Foundations of Attack–Defense Trees. In: Degano P, Etalle S, Guttman J, editors. Formal Aspects of Security and Trust: 7th International Workshop, FAST 2010, Pisa, Italy, September 16-17, 2010 Revised Selected Papers. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. p. 80-95.

[29] Poolsappasit N, Dewri R, Ray I. Dynamic Security Risk Management Using Bayesian Attack Graphs. IEEE Transactions on Dependable and Secure Computing. 2012;9(1):61-74

[30] Jonathan D. Weiss. A system security engineering process. In 14th Nat. Comp. Sec. Conf.,pages 572–581, 1991.

[31] Bistarelli S, Fioravanti F, Peretti P, editors. Defense trees for economic evaluation of security investments. First International Conference on Availability, Reliability and Security (ARES'06); 2006 20-22 April 2006.

[32] Phillips c, Painton Swiler l; A graph-based system for network-vulnerabilityanalysis. In Proceedings of the 1998 Workshop on New Security Paradigms (NSPW'98),pages 71–79, Charlottesville, Virginia, USA, September 1998.

[33] S. N. Foley, W. M. Fitzgerald: Management of Security Policy Configuration using a Semantic Threat Graph Approach, Journal of Computer Security (JCS), IOS Press, Volume 19, Number 3, 2011

[34] Matulevicius R. "Fundamentals of Secure System Modelling", Chapter 4 (unpublished draft), 2016

[35] Viveros P.: Analysis of the Cyber Attacks against ADS-B Perspective of Aviation Experts. Master thesis. University of Tartu. Estonia 2016.

[36] ben Othmane L, Ranchal R, Fernando R, Bhargava B, Bodden E. Incorporating attacker capabilities in risk estimation and mitigation. Computers & Security. 2015 6//;51:41-61.

[37] Alberts C, Dorofee A. Managing Information Security Risks: The OCTAVE Approach, Addison-Wesley Professional.2002.

[38] Mauw, S., Oostdijk, M.: Foundations of Attack Trees. In: Won, D., Kim, S. (eds.): ICISC 2005, LNCS 3935, 186–198. Springer (2005)

[39] Buldas, A., Laud, P., Priisalu, J., Saarepera, M., Willemson, J.: Rational choice of security measures via multi-parameter attack trees. In: López, J. (ed.):CRITIS 2006, LNCS 4347, 235–248. Springer (2006)

[40] Jürgenson, A., Willemson, J.: Computing Exact Outcomes of Multi-parameter Attack Trees. In: Meersman, R., Tari, Z. (eds.): OTM 2008, LNCS 5332, 1036–1051. Springer (2008)

[41] Jürgenson, A., Willemson, J.: Serial Model for Attack Tree Computations. In: Lee, D., Hong, S. (eds.): ICISC 2009, LNCS 5984, 118–128. Springer (2009)

[42] Buldas, A., Stepanenko, R.: Upper bounds for adversaries' utility in attack trees. In: Grossklags, J., Walrand, J.C. (eds.): GameSec 2012, LNCS 7638, 98–117. Springer (2012)

[43] Adam Shostack(2014). Threat Modeling Designing for Security-Appendix C. Wiley.

[44] Othmane, Lotfi ben, Ranchal, Rohit, Fernando, Ruchith, Bhargava, Bharat, & Bodden, Eric. (2015). Incorporating attacker capabilities in risk estimation and mitigation. Comput. Secur., 51(C), 41-61. doi: 10.1016/j.cose.2015.03.001

Appendix

I. Glossary

II. Requirements of the Implemented Aligned Tool

In this chapter we provide a complete picture over functional requirements and architecture of the tool which simulate the risk.

Textual Use case

The implemented Aligned ADTree should be able to perform CRUD operations on the main concepts of ISSRM like asset, impact, vulnerability, and attack methods. Here we explain each of these requirements using Use-Case diagrams

Table 43- Create an Asset

Use case Name	Create an Asset
Use Case ID	5
Requirement ID	3
Brief Description	1-Use case starts when the user choose the Asset in Edit menu 2-The user enters name, value, description of the asset, and number of attacks on the asset 3-The user select a vulnerability 4-The system registers the data into the database 5-The system provides a message regarding to success or failure of insertion into the asset table 6-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Asset* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of insertion into asset table.

Table 42-Edit Asset Usecase

Use case Name	Edit an Asset
Use Case ID	6
Requirement ID	3
Brief Description	1-Use case starts when the user choose the Asset in Edit menu, and then chooses the Edit asset tab. 2-The system loads all the asset data into a table 3-The user select a row (an asset) in table 4-The user clicks on update button

	5-The system opens a window which contains all current data of selected asset 6-User can update the loaded data 7-User clicks the update button 8-System provides a message regarding success or failure of update 8-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Asset* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of updating record in asset table.
Exception Flow	E1: No asset is selected 1-Use case starts when the user choose the Asset in Edit menu, and then chooses the Edit asset tab. 2-The system loads all the asset data into a table 3-The user clicks on update button 4-The system opens a dialogbox, and asks the user to select an asset.

Table 44-Delete asset usecase

Use case Name	Delete Asset
Use Case ID	**7**
Requirement ID	3
Brief Description	1-Use case starts when the user choose the Asset in Edit menu, and then chooses the Edit asset tab. 2-The system loads all the asset data into a table 3-The user select a row (an asset) in table 4-The user clicks on delete button 5-The system deletes the selected asset from Asset table in database 6-The system updates the table view on opened window 7-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Asset* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of updating record in asset table.
Exception Flow	E1: No asset is selected 1-Use case starts when the user choose the Asset in Edit menu, and then chooses the Edit asset tab. 2-The system loads all the asset data into a table 3-The user clicks on update button

	4-The system opens a dialogbox, and asks the user to select an asset.

Table 45-Add vulnerability usecase

Use case Name	Add a Vulnerability
Use Case ID	10
Requirement ID	2
Brief Description	1-Use case starts when the user choose the Vulnerability in Edit menu, and then chooses the Edit Vulnerability tab. 2-The system loads all the Vulnerability data into a table 3-The user select a row (an Vulnerability) in table 4-The user clicks on update button 5-The system opens a window which contains all current data of selected Vulnerability 6-User can update the loaded data 7-User clicks the update button 8-System provides a message regarding success or failure of update 8-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Vulnerability* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of insertion into vulnerability table.

Table 46-Edit vulnerability usecase

Use case Name	Edit Vulnerability
Use Case ID	11
Requirement ID	2
Brief Description	1-Use case starts when the user choose the Vulnerability in Edit menu, and then chooses the Edit Vulnerability tab. 2-The system loads all the Vulnerability data into a table 3-The user select a row (an Vulnerability) in table 4-The user clicks on Update button 5-The system deletes the selected asset from Vulnerability table in database 6-The system updates the table view on opened window 7-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses Vulnerability from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of updating record in asset table.

Exception Flow	E1: No asset is selected 1-Use case starts when the user choose the Vulnerability in Edit menu, and then chooses the Edit asset tab. 2-The system loads all the asset data into a table 3-The user clicks on delete button 4-The system opens a dialogbox, and asks the user to select a Vulnerability first, and try again

Table 47-Delete vulnerability usecase

Use case Name	Delete Vulnerability
Use Case ID	12
Requirement ID	3
Brief Description	1-Use case starts when the user choose the Vulnerability in Edit menu, and then chooses the Edit Vulnerability tab. 2-The system loads all the Vulnerability data into a table 3-The user select a row (an Vulnerability) in table
	4-The user clicks on Delete button 5-The system deletes the selected asset from Vulnerability table in database 6-The system updates the table view on opened window 7-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses Vulnerability from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of updating record in asset table.
Exception Flow	E1: No asset is selected 1-Use case starts when the user choose the Vulnerability in Edit menu, and then chooses the Edit asset tab. 2-The system loads all the asset data into a table 3-The user clicks on delete button 4-The system opens a dialogbox, and asks the user to select a **Vulnerability** first, and try again

Table 48-Create countermeasure usecase

Use case Name	Create Countermeasure
Use Case ID	19
Requirement ID	5
Brief Description	1-Use case starts when the user choose the Atomic Countermeasure in Edit menu 2-The user enters name, description, cost of implementation, and failure probability of the countermeasure 4-The system inserts the data into the database 5-The system provides a message regarding to success or failure of insertion into the countermeasure table 6-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses Atomic countermeasure from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of insertion into asset table.

Table 49- update countermeasure usecase

Use case Name	Update Countermeasure
Use Case ID	21
Requirement ID	5
Brief Description	1-Use case starts when the user choose the Atomic Countermeasure in Edit menu, and then chooses the Edit Atomic Countermeasure tab. 2-The system loads all the Countermeasure data into a table 3-The user select a row (an Countermeasure) in table 4-The user clicks on update button 5-The system opens a window which contains all current data of selected Countermeasure 6-User can update the loaded data 7-User clicks the update button 8-System provides a message regarding success or failure of update 8-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Countermeasure* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of updating record in impact table.
Exception Flow	E1: No impact is selected 1-Use case starts when the user choose the Atomic *Countermeasure* in Edit menu, and then chooses the Edit Atomic *Countermeasure* tab.

	2-The system loads all the *Countermeasure* data into a table 3-The user clicks on update button 4-The system opens a dialogbox, and asks the user to select an asset.

Table 50- Delete countermeasure usecase

Use case Name	Delete Countermeasure
Use Case ID	20
Requirement ID	5
Brief Description	1-Use case starts when the user choose the Atomic Countermeasure in Edit menu, and then chooses the Edit Atomic countermeasure 2-User clicks the delete button 8-System deletes the countermeasure and updates the table 9-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Atomic Countermeasure* from *Edit* menu
Post-condition	When use case is done, the table of countermeasures get updated
Exception Flow	E1: No impact is selected 1-Use case starts when the user choose the Atomic *Countermeasure* in Edit menu, and then chooses the Edit Atomic *Countermeasure* tab. 2-The system loads all the *countermeasures* data into a table 3-The user clicks on delete button 4-The system opens a dialogbox, and asks the user to select an asset.

Table 51-Assign an Attack to a Countermeasure Node Countermeasure

Use case Name	Assign an Attack to an Countermeasure Node
Use Case ID	24
Requirement ID	8
Brief Description	1-Use case starts when the user choose the new ADTree from File menu 2-The user right clicks on the root node 4-User chooses the add child from pop up menu 5-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses new ADTree from File menu
Post-condition	When use case is done, a child attack node is added to the root node

Table 52- Assign a Countermeasure Node to an Attack usecase

Use case Name	Assign a Countermeasure Node to an Attack
Use Case ID	23

Requirement ID	8
Brief Description	1-Use case starts when the user choose the new ADTree from File menu 2-The user right clicks on the root node 4-User chooses the *add countermeasure* from pop up menu 5-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses new ADTree from File menu
Post-condition	When use case is done, a child countermeasure node is added to the root node

Table 53- Assign an atomic attack to each attack node in leaves usecase

Use case Name	Assign an atomic attack to each attack node in leaves
Use Case ID	25
Requirement ID	7
Brief Description	1-Use case starts when the user choose the new ADTree from File menu, user adds some children to the root 2-The user right clicks on a leaf attack node 4-User chooses the ***assign an atomic action*** from pop up menu 5-System opens a window which lists all the atomic attack actions 6-User selects one atomic action 7-User clicks "Ok" 5-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses new ADTree from File menu. User adds some children to the root attack node
Post-condition	When use case is done, when the window with the table of atomic actions are closed

Table 54- Assign an atomic countermeasure to each countermeasure node in leaves usecase

Use case Name	Assign an atomic countermeasure to each countermeasure node in leaves
Use Case ID	26
Requirement ID	7
Brief Description	1-Use case starts when the user choose the new ADTree from File menu, user adds a countermeasure to an attack node. Then add some children to the countermeasure node. 2-The user right clicks on a leaf countermeasure node 4-User chooses the *assign an atomic action* from pop up menu 5-System opens a window which lists all the atomic countermeasure actions 6-User selects one atomic action

	7-User clicks “Ok” 5-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses new ADTree from File menu. User adds a countermeasure child to the root attack node.
Post-condition	When use case is done, when the window with the table of atomic actions are closed

Table 55- See properties of each node usecase

Use case Name	See properties of each node
Use Case ID	31
Requirement ID	8,9
Brief Description	1-Use case starts when the user choose the new ADTree from File menu, adds some nodes, and assigned atomic actions to all leaves. 2-The user right clicks on node 3-User chooses the *properties* from pop up menu 5-System computes costs of the node 6-System computes probability of the node 7-User selects one atomic action 8-System open a window which shows the cost, probability of the node 5-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses new ADTree from File menu. User adds a countermeasure child to the root attack node.
Post-condition	When use case is done, when the window with the table of atomic actions are closed

Table 56- Apply Threat Agent Profile on Properties of Attack Node usecase

Use case Name	Apply Threat Agent Profile on Properties of Attack Node
Use Case ID	35
Requirement ID	11
Brief Description	1-Use case starts when the user choose the new ADTree from File menu, adds some nodes, and assigned atomic actions to all leaves. 2-The user right clicks on node 3-User chooses the *properties* from pop up menu 5-System computes costs of the node 6-System computes probability of the node 7-User set up a profile for threat agent

	8-User clicks on the button 8-System applies the threat agent profile on node 5-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses new ADTree from File menu. User adds a countermeasure child to the root attack node.
Post-condition	When use case is done, when the window with the table of atomic actions are closed

Table 57- Choose the metric in disjunctive parents node usecase

Use case Name	Choose the metric in disjunctive parents node
Use Case ID	36
Requirement ID	13
Brief Description	1-Use case starts when the user runs the application 2-The system asks to choose the metric 3-User select one of the options 7-User clicks on the "Ok" button 5-Usecase ends
Pre-condition	The user runs the application
Post-condition	When use case is done, the main window of application is open

Table 58-Create impact usecase

Use case Name	Create an Impact
Use Case ID	1
Requirement ID	1
Brief Description	1-Use case starts when the user choose the Impact in Edit menu 2-The user enters name, and description of the impact 3-The user select a level of impact 3-The user select impact type 4-The system inserts the data into the database 5-The system provides a message regarding to success or failure of insertion into the impact table 6-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Impact* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of insertion into asset table.

Table 59-update impact usecase

Use case Name	Update Impacts
Use Case ID	2
Requirement ID	1
Brief Description	1-Use case starts when the user choose the Impact in Edit menu, and then chooses the Edit Impact tab. 2-The system loads all the impact data into a table 3-The user select a row (an impact) in table 4-The user clicks on update button 5-The system opens a window which contains all current data of selected asset 6-User can update the loaded data 7-User clicks the update button 8-System provides a message regarding success or failure of update 8-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Impact* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of updating record in impact table.
Exception Flow	E1: No impact is selected 1-Use case starts when the user choose the Impact in Edit menu, and then chooses the Edit Impact tab. 2-The system loads all the impact data into a table 3-The user clicks on update button 4-The system opens a dialogbox, and asks the user to select an asset.

Table 60-Delete impact usecase

Use case Name	Delete Impact
Use Case ID	3
Requirement ID	1
Brief Description	1-Use case starts when the user choose the Impact in Edit menu, and then chooses the Edit Impact tab. 2-The system loads all the asset data into a table 3-The user select a row (an impact) in table 4-The user clicks on delete button 5-The system deletes the selected impact from impact table in database 6-The system updates the table view on opened window 7-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Impact* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of updating record in asset table.

Exception Flow	E1: No impact is selected 1-Use case starts when the user choose the Impact in Edit menu, and then chooses the Edit impact tab. 2-The system loads all the asset data into a table 3-The user clicks on delete button 4-The system opens a dialogbox, and asks the user to select an impact.

Table 61-Delete attack usecase

Use case Name	Update Atomic Attack
Use Case ID	17
Requirement ID	4
Brief Description	1-Use case starts when the user choose the Atomic Attack in Edit menu, and then chooses the Edit Atomic Attack tab. 2-The system loads all the Atomic Attacks data into a table 3-The user select a row (an Attack) in table 4-The user clicks on update button 5-The system opens a window which contains all current data of selected Attack 6-User can update the loaded data 7-User clicks the update button 8-System provides a message regarding success or failure of update 8-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Atomic Attacks* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of updating record in impact table.
Exception Flow	E1: No impact is selected 1-Use case starts when the user choose the Atomic *Attack* in Edit menu, and then chooses the Edit Atomic *Attack* tab. 2-The system loads all the *Attacks* data into a table 3-The user clicks on update button 4-The system opens a dialogbox, and asks the user to select an asset.

Table 62-Delete attack usecase

Use case Name	Delete an Atomic Attack
Use Case ID	16
Requirement ID	4

Brief Description	1-Use case starts when the user choose the Atomic Attack in Edit menu, and then chooses the Edit Atomic Attack tab. 2-The system loads all the Atomic Attacks data into a table 3-The user select a row (an Attack) in table 4-The user clicks on update button 5-The system opens a window which contains all current data of selected Attack 6-User can update the loaded data 7-User clicks the update button 8-System provides a message regarding success or failure of update 9-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses *Atomic Attacks* from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of updating record in impact table.
Exception Flow	E1: No impact is selected 1-Use case starts when the user choose the Atomic *Attack* in Edit menu, and then chooses the Edit Atomic *Attack* tab. 2-The system loads all the *Attacks* data into a table 3-The user clicks on update button 4-The system opens a dialogbox, and asks the user to select an asset.

Table- create atomic attack usecase

Use case Name	Create an Atomic Attack
Use Case ID	15
Requirement ID	4
Brief Description	1-Use case starts when the user choose the Atomic Attacks in Edit menu 2-The user enters name, description, cost of attack, probability of success, and gain of attack 4-The system inserts the data into the database 5-The system provides a message regarding to success or failure of insertion into the attack table 6-Usecase ends
Pre-condition	The user has chosen the metric for evaluating of disjunctive refinements nodes. Then the user chooses Atomic Attacks from *Edit* menu
Post-condition	When use case is done, user will receive a message about success or failure of insertion into asset table.

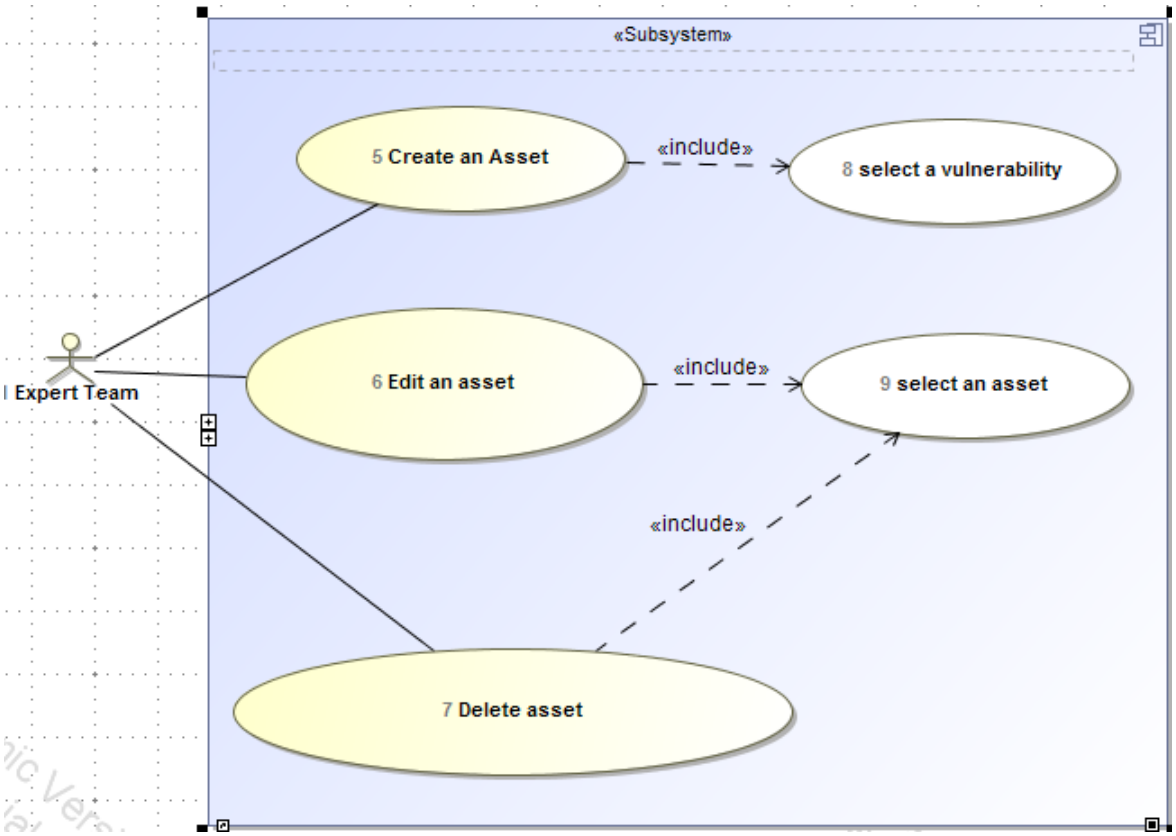


Fig 28. Asset Use cases

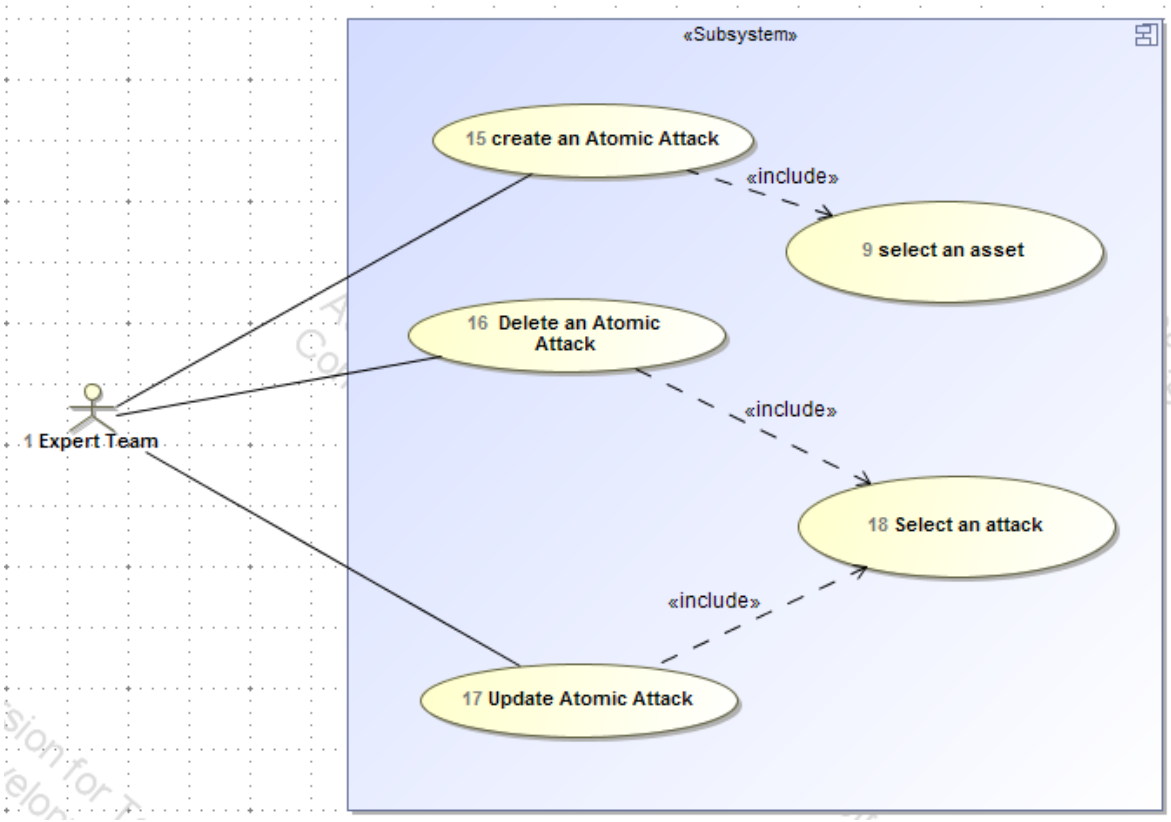


Fig 29. Attack Use cases

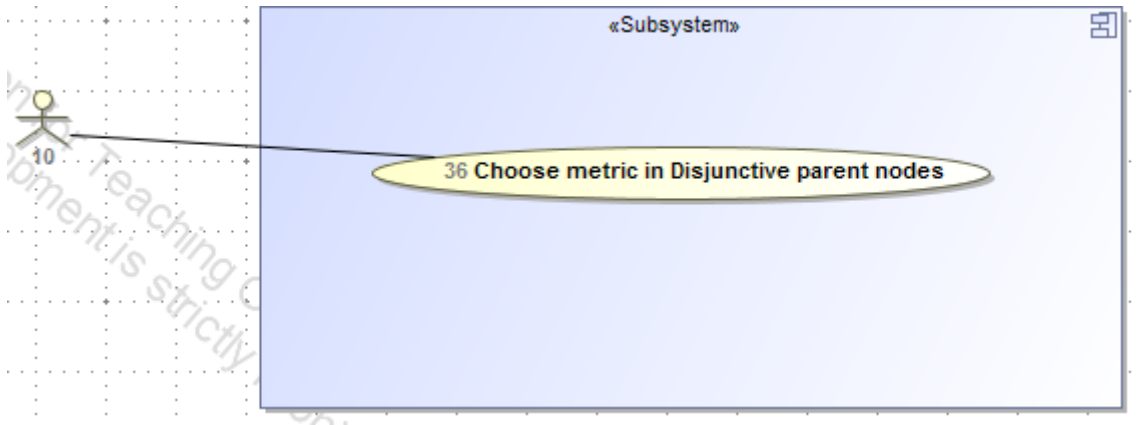


Fig 30. Decision in Disjunctive nodes

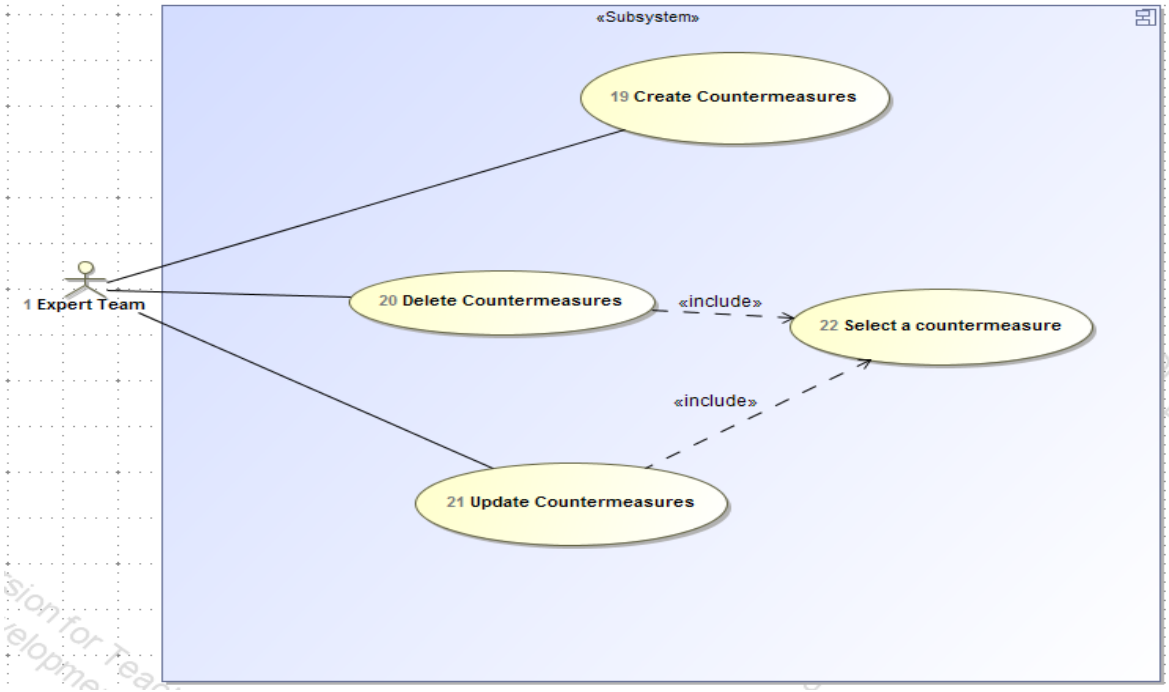


Fig 31. Countermeasure Use cases

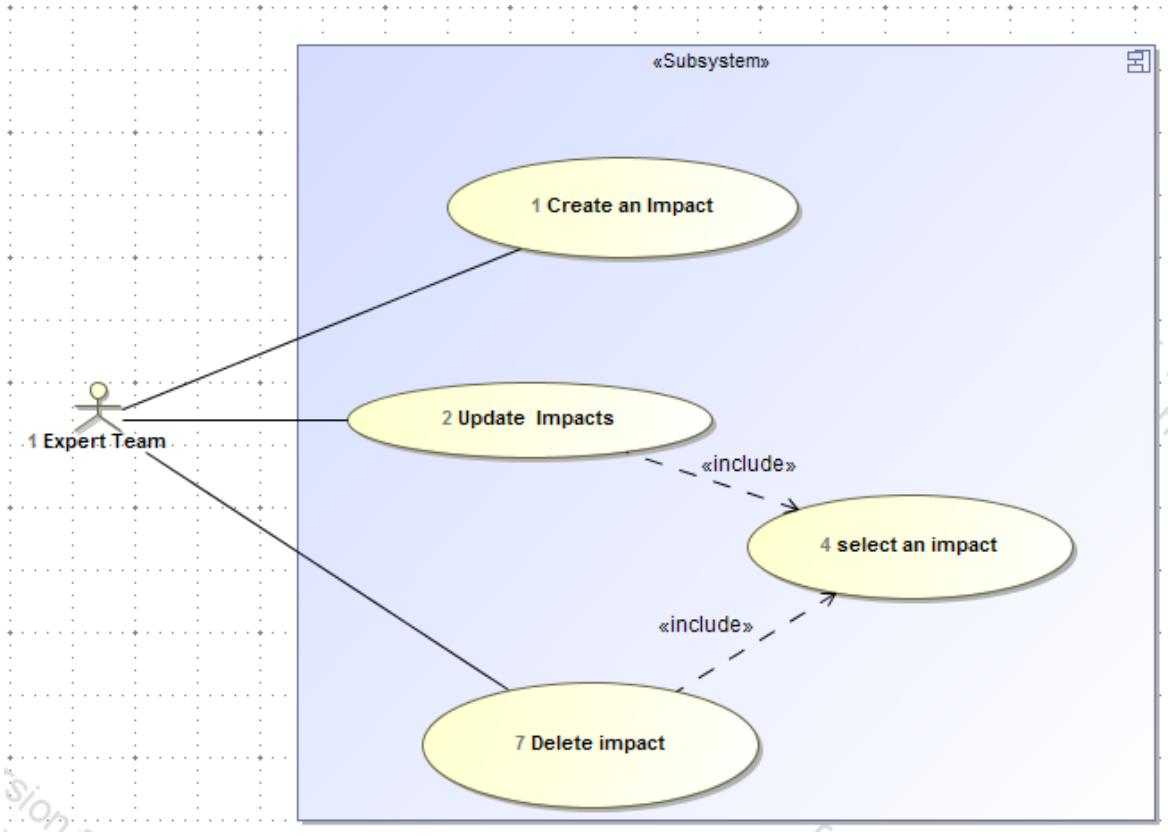


Fig 32. Decision in Disjunctive nodes

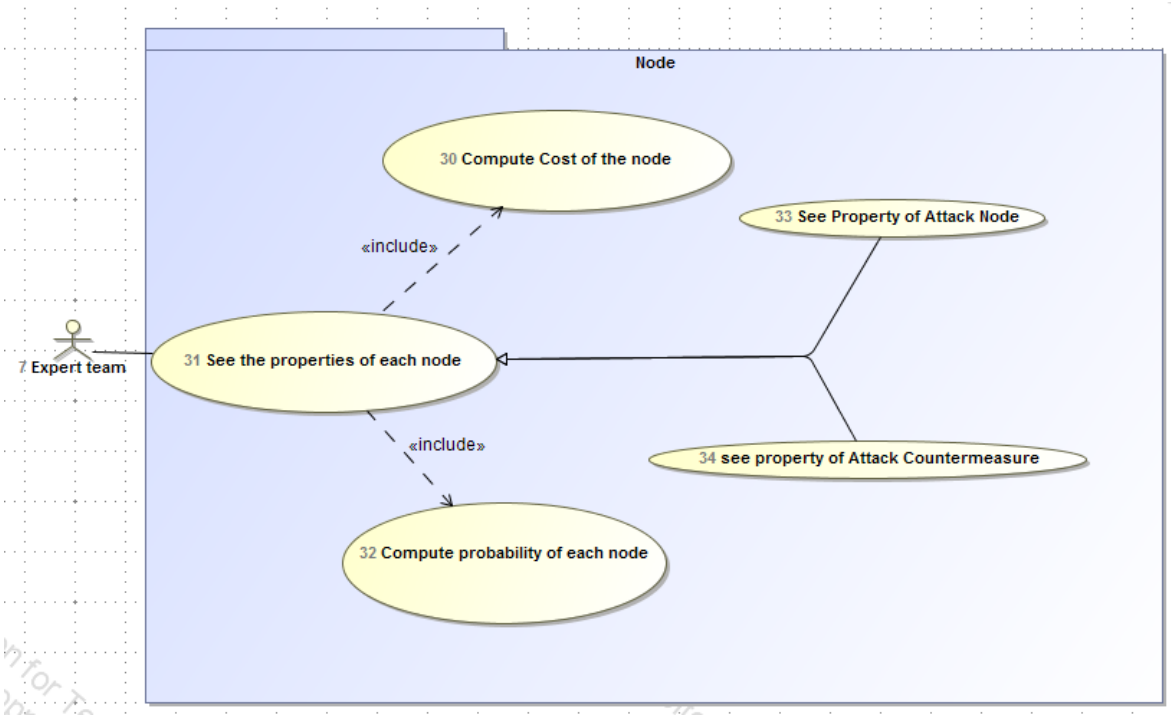


Fig 33. Node properties usecase

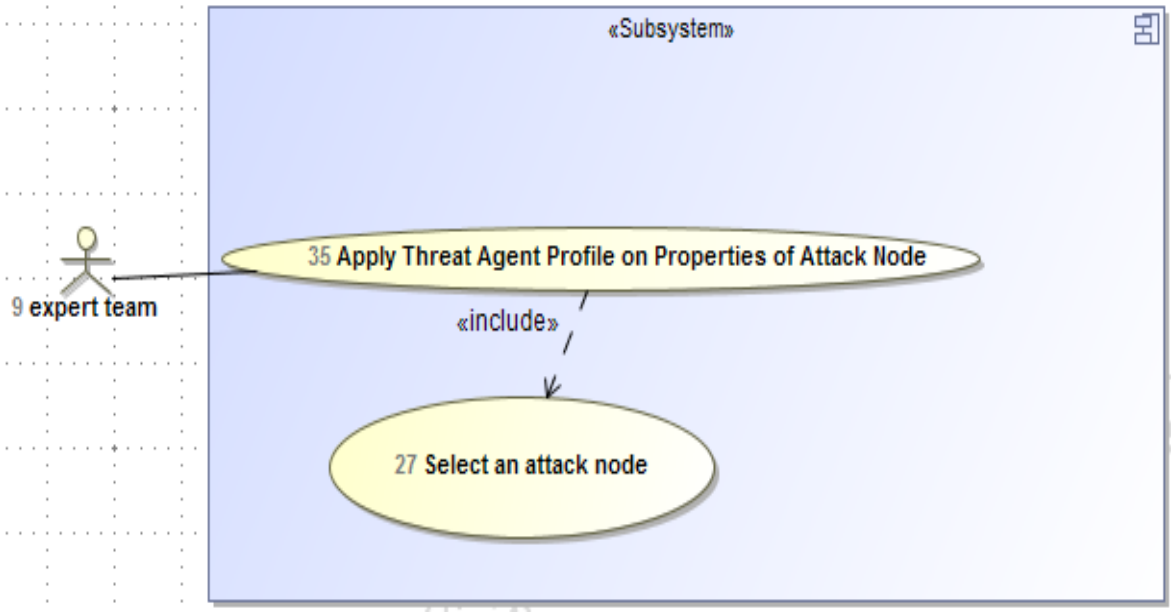


Fig 34. Apply threat agent profile on probability of attack

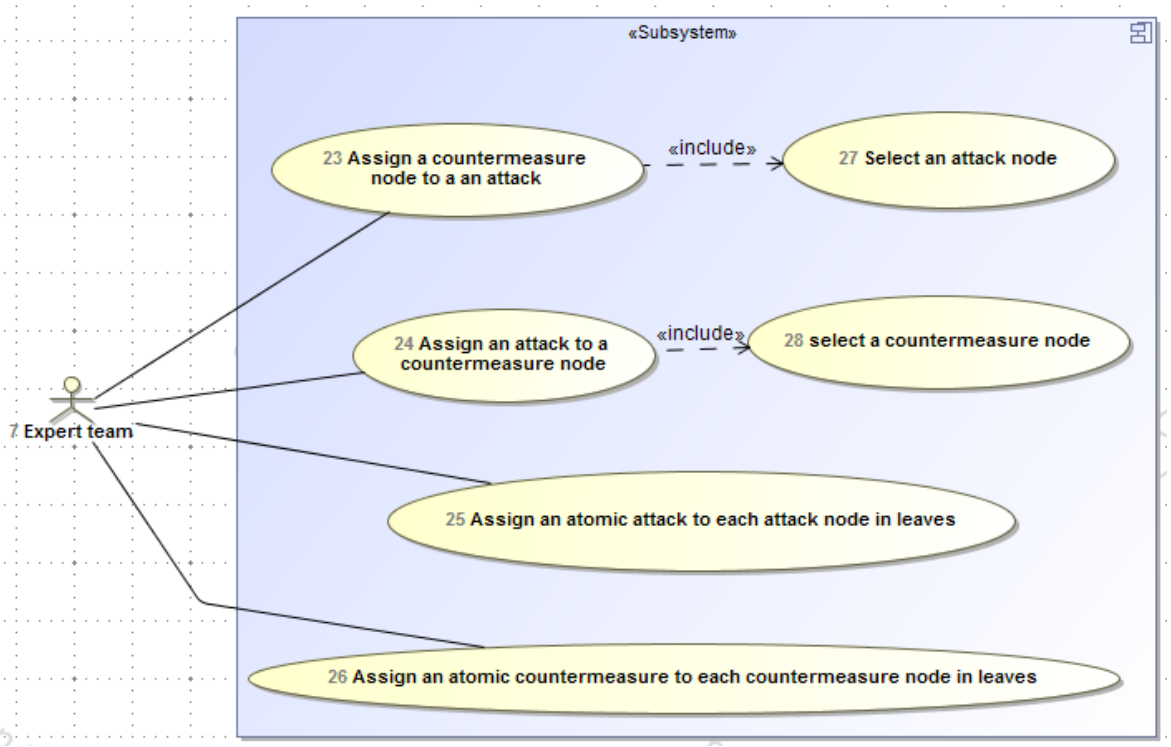


Fig 35. ADTree usecases

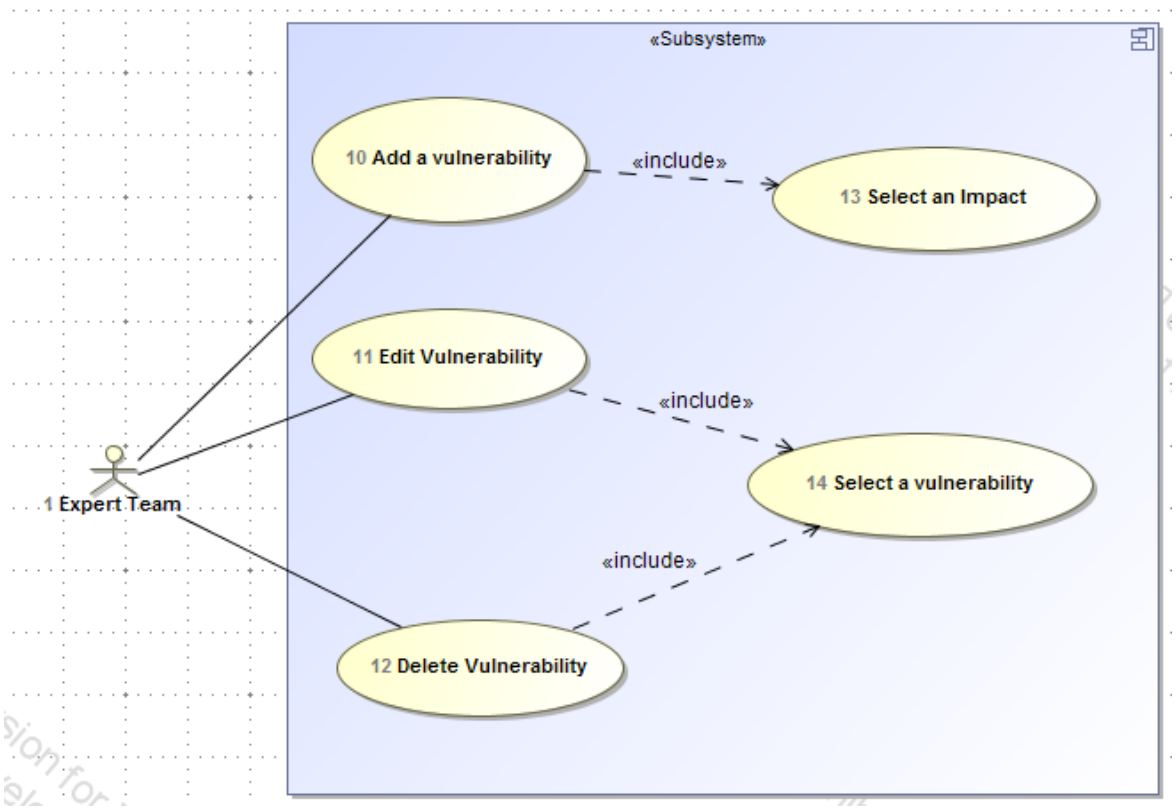


Fig 36. Vulnerability usecases

III. License

Non-exclusive license to reproduce thesis and make thesis public

I, **Salman Lashkarara**,

(**Salman Lashkarara**)

2.1 herewith grant the University of Tartu a free permit (non-exclusive license) to:

1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Managing Security Risks Using Attack-Defense Trees,

(title of thesis)

supervised by *Raimundas Matulevicius*,

(Raimundas Matulevicius)

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive license does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **12.05.2017**