# THE MOTIVATION OF ATTACKERS IN ATTACK TREE ANALYSIS

14 AUGUSTUS 2015

RICK VAN HOLSTEIJN - 4006356
Supervisors:
Prof.dr.ir. J. van den Berg
Dr.ir. W. Pieters
Dr. M.P.M. Franssen
Prof.dr.ir. P.H.A.J.M. van Gelder

# Executive summary

The number of cyberattacks has been growing over time and is expected to keep growing. In order to prevent such attacks, countermeasures have to be put in place by IT security experts. These IT security experts are however often tied to budgets and do not have a good overview of the threats that are present. It is thus necessary to provide them tools that can help them to decide on how to allocate their resources. One of these tools is the attack tree methodology, which is used to analyse complex attacks that consist of multiple steps. Properties of the overall security of the system are derived by properties of the smaller steps.

These properties of the attack are represented in the form of parameters that are allocated to the nodes in the attack tree. Some of these parameters used are however dependent on the type of attacker. In order to be able to reuse the attack tree for analysing it for various types of attackers, the parameters in the attack tree have to be made independent of the attacker. In order to do so, attacker properties are considered separately, which are summarized in attacker profiles. So far, methods have been formed to include the attacker's resources and the attacker's skill in the attack tree methodology. The result of the current research is a framework that includes the motivation of the attacker in the attack tree methodology. The framework can be used by IT security experts to analyse the attack tree for variously motivated attackers, without having to update the parameter values.

A design science approach is used to design the framework, which starts with the identification of the knowledge gap. The knowledge gap lies in how to include the motivation of the attacker in the attack tree methodology. This motivation is assumed to have an influence on the pay-off an attacker receives from performing an attack. The value that including the motivation in the analysis can bring can be summarized as the following:

- The gains parameter is made independent of the type of attacker
- Various pay-offs are possible for variously motivated attackers
- The gains parameter is made more realistic

The framework is ensured to reach this potential added value, by adhering to a list of requirements. This list of requirements is build up from constraints to which the framework must conform and dilemmas for which a design choice has to be made. The resulting framework is mainly based on the method presented by Lenin et al. (2014). Changes to the current method are mainly made to the gains parameter. The gains is no longer a global parameter that is only received by the attacker when reaching the root node. Instead it is possible to include intermediate pay-offs, which means that gains can also be allocated to intermediate nodes. In this way, different gains are possible for different attack paths in the attack tree. The gains can thus be represented in a more fine grained way. Also an opt-out possibility is included to allow attackers to perform attacks to only reach an intermediate node and not the root node of the attack tree.

In the current method the pay-off for an attacker was considered to be equal to the gains, which was the same for every type of attacker. This gains was also a single value. In the designed framework the gains has been slit up in five types of gains to deal with the five forms of motivation that an attacker may have, which are financial benefits, causing damage, knowledge gaining, pleasure seeking and gaining notoriety within a community. With the use of weight values, the importance of the various types of gains for an attacker can be represented. By multiplying the gains with the weight values, a pay-off can be calculated for a certain type of attacker. This way various pay-offs are possible for variously motivated attackers.

A case study has been described to show the working of the framework on a real world case, which also served as a validation of the framework. In addition an expert opinion has been asked to validate the framework. The main improvement that can be made to the framework by future research is focussed on allocating values to the different types of gains and allocating the weight values for the different types of gains. Also attention could be paid to several dependencies between attack and attacker properties that have not yet been taken into account.

# Acknowledgments

After a full semester of research, this thesis is the result that would not have been in this state if it wasn't for the help of some people. First of all I would like to thank my graduation committee consisting of Prof.dr.ir. Jan van den Berg, Dr.ir. Wolter Pieters, Dr. Maarten Franssen and Prof.dr.ir. Pieter van Gelder for the valuable feedback and for providing me the necessary guidance in performing this research. Especially I would like to thank Wolter for the intensive guidance during the research. He was also the one to bring me in touch with Jan Willemson and Aleksandr Lenin from Cybernetica. The help of both Jan and Aleksandr has been very useful in forming this thesis, for which I would thus like to thank them. Also I would like to thank Barbara Kordy and Marielle Stoelinga for the feedback given on my thesis.

My thanks go out to Dr.ir. Dina Hadžiosmanović and Wolter for forming the TREsPASS master students group, which was very useful. I would like to thank Katrien Meijdam, Demetris Antoniou, Harikrishnan Pushpakumar and Yiwen Zhu for the valuable feedback provided to me during the TREsPASS master student meetings and in other instances.

Last but not least I would like to thank my girlfriend, my parents and my family and friends for supporting me in the process of forming this thesis. Discussing my work with them has helped me further in the research. Also it was possible to turn to them if it was necessary to put my mind to something else for a moment.

Rick van Holsteijn, 's-Gravenzande 2015

# List of definitions

| Term | Definition |
|---|---|
| **Attacker** | An attacker is a person that attempts to harm a system by performing an attack on it |
| **Attacker profile** | An attacker profile is a description of a certain type of attacker in which various characteristics are included |
| **Attack tree** | An attack tree is a visualisation method used in the attack tree methodology, where attacks are represented in a tree like structure. This attack tree is used as the basis for the analysis in the attack tree methodology |
| **Elementary attack** | An elementary attack is an attack that is considered to be simple enough to easily assign parameters to it |
| **Intermediate node** | An intermediate node is a node in the attack tree that is not the root node and has two or more child nodes |
| **Leaf node** | A leaf node is a node in the attack tree that has no child nodes and has one or more parent nodes |
| **Root node** | The root node is the top node in an attack tree, which represents the overall attack that the attacker is trying to perform |

# Content

# List of figures and tables

# 1 Introduction

In recent years, the world has become more and more connected via the internet. Although this has brought great convenience and many new technological opportunities, it also brought along new possibilities for criminals in the form of cybercrime. This called for the need of cybersecurity, which is defined by the Merriam-Webster dictionary as “measures taken to protect a computer or computer system (as on the internet) against unauthorized access or attack”. Over time the number of attacks performed has increased tremendously and attacks are becoming more and more complex (Van Kessel & Allan, 2013). It is expected that the amount of attacks will continue to multiply (Van Kessel & Allan, 2014). There are estimates that say that in 2013 there were more than 800 million incidents (McAfee, 2014). The economic impact of cybercrime is estimated to $400 billion per year by McAfee (2014). These numbers emphasize the need for effective cybersecurity efforts.

One of the problems is that IT security experts are often tied to strict budgets (Van Kessel & Allan, 2014), which means that decisions will have to be made on which measures to take. Also the IT security experts often do not have a very good insight in the cyber threats that exist (Van Kellel & Allan, 2014). In order to help IT security experts to take the right decisions and to give them more insight in the cyber threats they are dealing with, tools have been developed. One of these tools is the attack tree methodology which is the focus of this research.

### Introducing the attack tree methodology

The attack tree methodology was founded by Weiss (1991), who introduced a system security engineering process to help allocate resources to vulnerabilities with the highest risk. In his method he uses a tree like structure to decompose threats, which he names threat logic trees. The name attack tree was first used by Schneier (1999) who described the process of setting up an attack tree and allocating parameters to the attack tree to answer various questions about the security of a system. A possible question would for example be whether it is possible for an attacker to perform a certain attack. Also the attack tree could be used to find the least costly attack path for an attacker. An example visualisation of an attack tree is presented in Figure 1. The root node represents the primary threat which is the main objective for the attacker. This attack is then decomposed in sub attacks. Every node that is decomposed can either be an AND node or an OR node. For an AND node all the lower level objectives need to be performed successfully in order to reach the objective in the AND node. If for an OR node any of the lower level objectives is performed successfully, the objective in the OR node is reached (Weiss, 1991). Nodes that need no further decomposition are called elementary attacks, which are the leaf nodes of the attack tree. Section 3 further explains the use of attack trees.

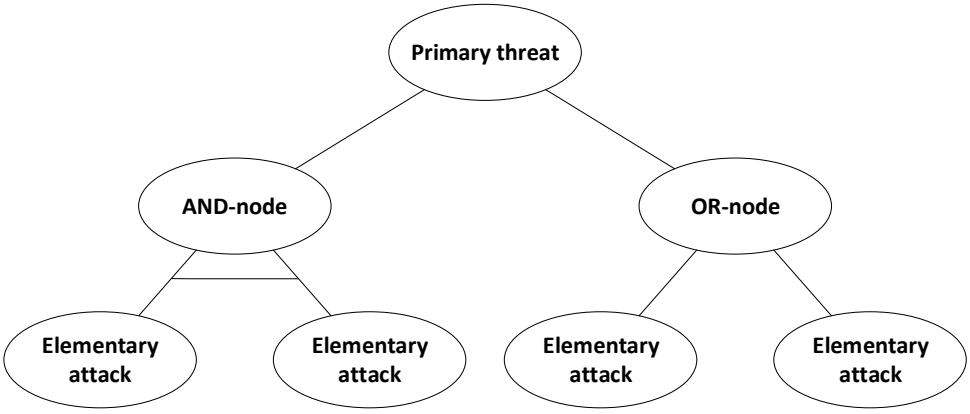


*Figure 1: Example visualization of an attack tree*

By using the attack tree structure, overall security properties can be derived from the properties of smaller steps (Mauw & Oostdijk). This makes the methodology useful for analysing complex attacks

that consist of multiple steps and where alternative paths are possible (Pieters & Davarynejad, 2014). The parameters that are allocated to the attack tree are often dependent on the attacker. For the reuse of an attack tree for several types of attackers it would however be better if the parameters that are assigned are attacker independent. In order to do so, the characteristics of the attacker need to be considered separately in the attack tree methodology. Recent research has put effort in separating the attacker properties from the system properties (Lenin et al., 2014).

Attacker profiles are used to summarize the properties of a certain attacker. By combining attacker profiles and attack trees that contain attacker independent parameters, the security of a system can be analysed for different types of attacker, without having to update the parameters of the attack tree. So far the resources and the skill of the attacker are the attacker properties that have been used in the attack tree methodology. Another important attacker property is the motivation that an attacker has for attacking the system, which has not yet been included in the attack tree methodology. The motivation of an attacker is interesting to take into account, because it may influence the way in which an attacker performs an attack. An attacker that is looking for fame would for example go for an attack that is more noticeable for outsiders than an attacker that is only interested in the financial benefits of performing an attack. This research describes the design of a framework for the attack tree methodology that does include the motivation of the attacker. With the help of this framework the security of the system can be analysed for variously motivated attackers.

### Stakeholders: The attacker and the defender

With the use of the attack tree methodology, attacks can be seen as games between the attacker and the defender, the IT security expert (Buldas et al., 2006). These two stakeholders have diametrically opposed interests. As the attacker tries to perform an attack with an as high as possible profit, the defender attempts to make sure no attacks are successfully performed on the system.

Attackers are often assumed to behave rational, which means that they will only attack when an attack has a positive expected outcome (Buldas et al., 2006). For the defender it is thus the goal to make sure there are no profitable attacks possible on the system. The attack tree methodology can be used to identify possible profitable attacks, which would thus help the IT security expert to determine where to take countermeasures. It can in turn also be used by the attacker to identify the attacks that have the highest expected outcome, which could help them decide what attack to perform.

### Research context: The TREsPASS project

Next to the stakeholders within the attack tree methodododology, there are also some stakeholders involved in the context of the research. First of all this research takes place in the light of the TREsPASS project. The project team aims to build an attack navigator that can predict, prioritise and prevent attack opportunities (TREsPASS, 2015). For this, knowledge is used from state-of-the-art industry processes and tools. The attack tree methodology is one of these tools that is used within the TREsPASS project. The better the attack tree methodology can thus predict or prioritise the possible attacks opportunities, the more useful it gets for the TREsPASS project team.

From the TREsPASS project team, Cybernetica is a company that has done various studies for setting up and improving the attack tree methodology (Buldas et al., 2006; Jürgenson & Willemson, 2007, 2008, 2010a, 2010b; Lenin et al., 2014). Cybernetica is the partner of the TREsPASS project team that was most involved in this research. The framework that is designed is thus based on the previously introduced method by researchers from Cybernetica.

### Objective and relevance of the research

The objective of this research is to design a framework in which the motivation of the attacker is included in the attack tree methodology. By doing so the attack tree methodology can be used to analyse the security of a system for different types of attackers in terms of what motivates them to

perform an attack. This will provide the IT security expert with a better insight in possible attacks that may be performed.

The social relevance of the research is tightly connected to this. If an IT security expert has a better overview of the possible attacks, he is able to make better decisions on what countermeasures to take, which should in turn lead to a higher security level. Already it was indicated that many incidents are occurring and these incidents do not only have companies as a target, but also individuals. More and more devices are connected to the internet and what is connected to the internet can actually be attacked (Poremba, 2015). This stresses the social relevance of good cybersecurity practices.

The research is also scientifically relevant, because of the current absence of a way to include the motivation of attackers in the attack tree methodology. Lenin et al. (2014) developed a method to include the skill and the resources of the attacker in the attack tree methodology, but the inclusion of the motivation of attackers is still a gap that needs to be resolved. The framework that is designed by means of the described research resolves this knowledge gap.

### Approach used and the result

The objective of the research is to design an extended framework for the attack tree methodology. The research is therefore a design oriented one. The approach used is based on the Design Science Research Methodology developed by Peffers et al. (2007). Section 2 further elaborates on the approach used. Five design steps have been performed to get to the framework that results from this research. In the resulted framework the IT security expert has more flexibility in representing the pay-offs for an attacker as compared to the current methods. In addition the IT security expert is able to calculate various pay-offs for variously motivated attackers without having to update the parameters in the attack tree. The framework has been validated by means of a case study and expert validation.

### Reading guide

This report is mainly structured by the steps that were performed to design the framework. First the approach used in this research is explained in more detail in section 2. In section 3 the current state of the art in the field of attack trees, attacker profiles and the combination of the two is described, which results in a description of the knowledge gap that is resolved within this research. This is followed by a description of the prospected added value of resolving this gap in section 4. Section 5 lists the requirements for the design of the extended framework for the attack tree methodology. The next step is to actually design the framework of which a description in given in section 6. The validation of the framework is discussed in section 7, which is followed by the conclusions in section 8. This report ends with a discussion of possible drawbacks of the framework and future research possibilities in section 9.

# 2 Research approach

This chapter explains the approach that was used for the research described in this report. The approach used is based on the design science research methodology developed by Peffers, Tuunanen, Rothenberger & Chatterjee (2007). First in section 2.1 the design science research methodology is described and it is argued why this approach is applicable to the current research. Section 2.2 lists the steps that are performed in the current research and the research questions that are connected to these steps. Finally section 2.3 presents a visualization of the research process and discusses the alterations that were made to fit the original research process to the current research.

## 2.1 The design science research methodology

As described in the introduction, the objective of this research is to extend the attack tree methodology by making it possible to analyse the attack tree for variously motivated attackers. In order to extend the attack tree methodology a new framework has to be designed. The approach that is used for the research is therefore a design oriented one. The problem defined is an IT related one, which is the reason for choosing a design approach from the information systems research. In the design science in information systems the focus lies on creating and evaluating artefacts (Hevner, 2004). These artefacts can take many forms under which also methods are mentioned, which is the type of artefact designed in the current research.

In his research essay Hevner (2004) describes guidelines for design science in information systems research, but no clear design process is presented. Based on these guidelines and various other studies into design science from other fields, Peffers et al. (2007) developed a general process for performing design science in information systems research, referred to as the Design Science Research Methodology (DSRM) Process Model. This model consists of six different steps to be performed. A visualization of the DSRM Process Model is shown in Figure 2. The italic text in each of the boxes gives the goal of that step.

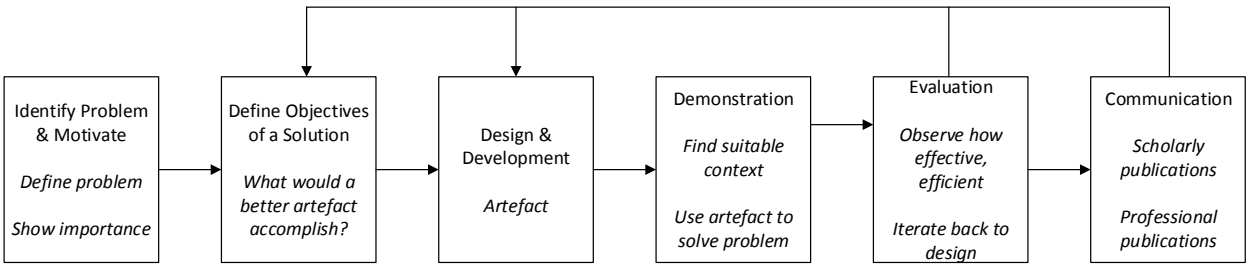


*Figure 2: DSRM Process Model (Based on Peffers et al., 2007)*

In his guidelines Hevner (2004) stressed that design science should be focussed on solving a relevant problem. The first step of the process is thus identifying the problem. Also a motivation should be given as for why the problem is a problem and why it needs to be solved. This is included in the first step of the process (Peffers et al., 2007).

The second step of the DSRM process is the defining of objectives for a solution. In this step the aim is to define requirements for the artefact to be designed. Based on knowledge of what is possible and feasible, objectives are set for the design (Peffers et al., 2007).

After defining the objectives and requirements, the actual designing of the artefact is the next step. This step is the central step of the DSRM Process Model, in which you determine the desired functionality of the design and then create the actual design (Peffers et al., 2007).

The two steps that follow are somewhat connected. In the demonstration step you show that the designed artefact actually solves the problem and in the evaluation step you check how well the artefact solves the problem (Peffers et al., 2007).

The last step of the DSRM process is the communication step, where the designer spreads the knowledge on the found solution and its effectiveness (Peffers et al., 2007). The arrows on top of the DSRM Process Model indicate that iterations are possible.

## 2.2 Steps to be performed and research questions

This section describes how the various steps from the DSRM process are performed in this research. Also research questions have been set up to guide the process. A main question is formulated to clearly describe the aim of this research. As described before, the research is design oriented, which is the reason that the main research question is design oriented. This main question is as follows:

*How can the motivation of attackers be included in the use of attack trees for cyber threat analysis?*

In order to find an answer to this main question, sub questions have been set up that all partly add up to the answer to the main question. The following sections describe these sub questions and the steps of the DSRM process that they are tied to. Also the way in which an answer to the sub questions is formed, is described.

### 2.2.1 Problem identification and motivation

The problem is already partly described in the introduction of this report, but the literature is studied to identify a well-defined knowledge gap to resolve with the help of this research. This is necessary to make sure that the problem is actually a real problem and to gain insight in the problem's details and complexity. The sub question tied to this is the following:

*$Q_1$: What is the current state of the art regarding:*

- *attack trees?*
- *attacker profiling, with a special focus on motivation?*
- *the combination of attacker profiles and attack trees?*

Answering this question results in the knowledge gap/problem for which the eventual design forms the solution. The answer to this question is found by means of an extensive literature study. In this first step of the DSRM process also the motivation for solving the problem should be given. Therefore there is another sub question tied to this step. The focus of the second sub question lies on the motivation for solving the research gap and is defined as follows:

*$Q_2$: What value could the inclusion of motivation in the use of attack trees add to the information gained from the attack tree analysis?*

Answering this question gives an overview of what solving the knowledge gap/problem can potentially add to the use of attack trees. For answering this question, also the literature is used.

### 2.2.2 Objective definition

The second step, the objective definition, is used to form requirements to which the solution to the problem needs to conform. These requirements make sure the potential added value is actually reached. Peffers et al. (2007) seem to use the terms objectives and requirements interchangeably. In this research the term requirements is used to cover both. These requirements are split into constraints and dilemmas. The constraints describe the requirements that the design has to comply with. The dilemmas describe the design choices for which a decision has to be made during the design and development step. The sub question that is related to the objective definition step is the following:

*$Q_3$: What are the requirements for a framework that includes the motivation of attackers in the use of attack trees?*

Answering this question results in a list of the requirements to which the design needs to conform. The results of the analysis performed in the first step are used to set up the requirements. This step is thus mainly based on literature research as well.

### 2.2.3 Design and development

The first phase of the design and development step is to determine the way in which you want the artefact to function (Peffers et al., 2007). In this research, this is done by making design choices for each of the dilemmas defined. After making these choices, the actual design is formed, which is a framework for an extended attack tree methodology. Because the artefact is a framework, or method, demonstration is needed to clarify the design. This is done by means of an example use of the extended attack tree methodology. The demonstration in the form of demonstrating the use of the artefact is thus also included in this step.

Because the design and development step is such a central part of the design, the sub question is closely related to the main research question. The requirements set up are however taken into account. The sub question is formulated as follows:

*$Q_4$: How to include the motivation of attackers in the use of attack trees with regard to the requirements?*

The answer to this question is the description of the design that solves the knowledge gap/problem and a demonstration of its use. The design is formed by taking design decisions based on argumentation that is formed with the help of the analysis performed in the first two steps.

### 2.2.4 Evaluation

The next step in the process developed by (Peffers et al., 2007) would be demonstration, but part of that is already included in the design and development step. Also in the evaluation step, a demonstration is necessary, which has led to the decision to not consider demonstration as a separate step. The next step in this research process is thus the evaluation step, which can be done in various ways. The sub question tied to this step of the research process is the following:

*$Q_5$: Does the method add the expected value?*

Answering this question, makes sure that the design reaches the objective that was originally set (solving the research gap/problem). Hevner (2004) describes some of the possible techniques to be used for evaluation, of which two are used in this research. The first is a case study, where the design is studied in more depth by applying it to a real world case. An extensive attack tree on I-voting has been provided by Cybernetica, which is used for this case study. It is evaluated whether the framework for the extended attack tree methodology adds the expected value as described in the first step of the research process. The second technique that is used to evaluate the design is the informed argument. With this technique you build a convincing argument for the added value of the design (Hevner, 2004). In this research this is done by consulting experts to validate the design.

### 2.2.5 Communication

The last step of the design science methodology is the communication step. This report and the associated academic paper serve as the means for this step. No sub question has been formulated for this step. The report and the academic paper are considered to formulate an answer to the main research question.

## 2.3 Research process

The research process described above is summarized in Figure 3. In this diagram the red boxes represent the steps that are performed, the green boxes represent the methods used for the steps and the yellow boxes represent the results of performing the steps. In brackets also the (sub) questions answered in the steps are presented. The main difference from the DSRM Process Model developed by Peffers et al. (2007) is the absence of the demonstration step. This step is considered to be partly performed in the design and development step and the evaluation step and is therefore not included as a separate step.

Just as in the original model, the arrows at the top indicate that the research process in an iterative one. This report describes the process as a waterfall process, but during the actual design process, several iterations were made.

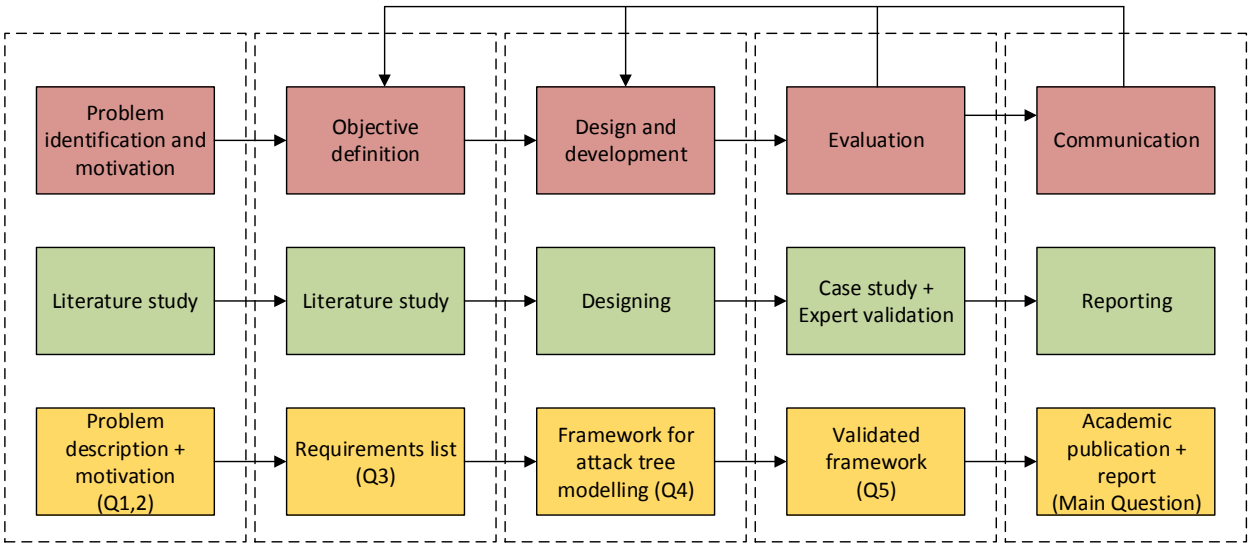


*Figure 3: Research process diagram based on DSRM Process Model (Peffers et al., 2007)*

# 3 The current state of the art: Attack trees and Attacker profiles

This chapter describes the body of knowledge in the field of cybersecurity on attack trees and attacker profiles. A description is given of what these two concepts entail and how the concepts have been developed historically. Also some special attention is paid to the interrelatedness of the two concepts and what this means for the overall project. The chapter gives an answer to the first sub question, *what is the current state of the art regarding:*

- *attack trees?*
- *attacker profiles, with a special focus on motivation?*
- *the combination of attack trees and attacker profiles?*

The chapter consists of four sections of which the first, 3.1, focusses on what attack trees are and how they are used. Section 3.2 describes the use of attacker profiles to describe attackers with various characteristics. After describing both of these concepts, section 3.3 focusses on the combination of the two. The chapter concludes with a description of the knowledge gap that was found in literature, which is described in section 3.4.

## 3.1 Attack trees

This section gives an elaborate description of attack trees, where attention is paid to what attack trees are, how the concept originated and what attack trees are used for. The attack tree methodology can be used by companies to describe possible attacks on their assets and to decide on what countermeasures to take in order to prevent these attacks from happening (successfully). First, the historic background of this methodology is given in 3.1.1, and after that the application of the methodology is explained in 3.1.2. In 3.1.3 some extensions of the original use of attack trees are discussed.

### 3.1.1 Historic background of attack trees

In a time where there was still a lot of scepticism towards spending design resources on security, Weiss (1991) introduced a system security engineering process to help allocate resources to vulnerabilities with the highest risk. In this process the vulnerabilities are decomposed with the help of a structuring method based on threat logic trees. In the root node of the tree an overall objective for an attacker is presented which is then decomposed in alternative objectives to reach this overall objective. Each of the nodes is either an OR node or an AND node. For an OR node any of the lower level objectives needs to be performed successfully in order to reach the objective in the OR node. For an AND node all of the lower level objectives need to be performed successfully in order to reach the objective in the AND node (Weiss, 1991).

After constructing the tree, parameters need to be allocated to the leaves in order to assess the risks associated with the threats. This phase is called the analysis phase. Weiss (1991) indicates that it is difficult to use the traditional formulas used in risk management based on the probability of occurrence and the impact of a threat. The reason he presents for this is that the probability of occurrence is often impossible to estimate, because this is too much dependent on the attacker. In order to estimate the risk, Weiss (1991) uses two parameters, of which one is based on the impact of the threat and the other is based on the resources required by an intelligent attacker to successfully execute the threat. The author does not provide a clear way in which values can be allocated for these two parameters. He does however state some elements that the parameters are build up from. Values for the two parameters are allocated to the leaf nodes, after which the risk is calculated. For the objectives in the intermediate nodes and the overall objective in the root node, the parameters are calculated based on their child nodes/leaves. After calculating the risk for each of the vulnerabilities an ordering is made based on these levels of risk. This way a ranking is formed of the vulnerabilities, that shows with which vulnerabilities the highest risk is associated.

The term attack trees was introduced a couple of years later by Schneier (1999), for a method that seems to be similar to the tree like structure used by Weiss (1991). Schneier describes attack trees as "*a formal, methodical way of describing the security of systems, based on varying attacks"* (Schneier, 1999). In the analysis phase of the attack tree, Schneier (1999) assigns parameters to the leaf nodes and uses these to determine the parameter values of the internal nodes and the root node. The most basic parameter he proposes is a Boolean parameter that can take the value 'possible' or 'impossible'. For an OR node, only one of the child node would need the value 'possible' for its value to be 'possible' as well. For an AND node, each of the child nodes would need the value 'possible'. He also states that other Boolean values can possibly be assigned or even continuous values. The key point of Schneier (1999) is that you determine what parameters to assign to the nodes in the attack tree based on the question that you would like to answer using the attack tree. For example, if you would want to know the cheapest possible way to reach the objective in the root node, you would assign a cost parameter to each of the nodes.

A formalization of the attack tree method was however still missing. Mauw & Oostdijk (2006) have provided a formalization of the concepts introduced by Weiss and Schneier. They indicate that this formalization is necessary for being able to build support tools for attack tree analysis. The formalization is based on three central aspects; attack suites, attacks and attack components. An attack suite is a set of attack components, which are represented in the leaf nodes of the attack tree. Each of these attack suites individually is called an attack. These definitions as introduced by Mauw & Oostdijk (2006) are used throughout the report. Also the mathematic structure that is used for the analysis phase of the attack trees is based on the formalization. In the next section this is further explained with the help of examples.

### 3.1.2 Application of attack trees

In the most basic use of attack trees, one tree is formed and parameters are assigned to the leaf nodes. The values of those parameters for the higher level nodes are derived from their child nodes. After doing so, an analysis is performed to decide what to focus the countermeasures on. In order to illustrate this some more, an example is worked out. This example is based on a simplified situation in which an attacker is aiming to obtain secret data from a company. This secret data can be obtained in two different ways, being through stealing a laptop OR by gaining remote access. In order to steal a laptop, a key needs to be socially engineered AND the room in which the laptop lies, needs to be accessed. Remote access can be obtained by cracking the password OR exploiting a vulnerability. The attack tree representing this situation is shown in Figure 4. The link under the 'Steal laptop' node indicates that it is an AND node.

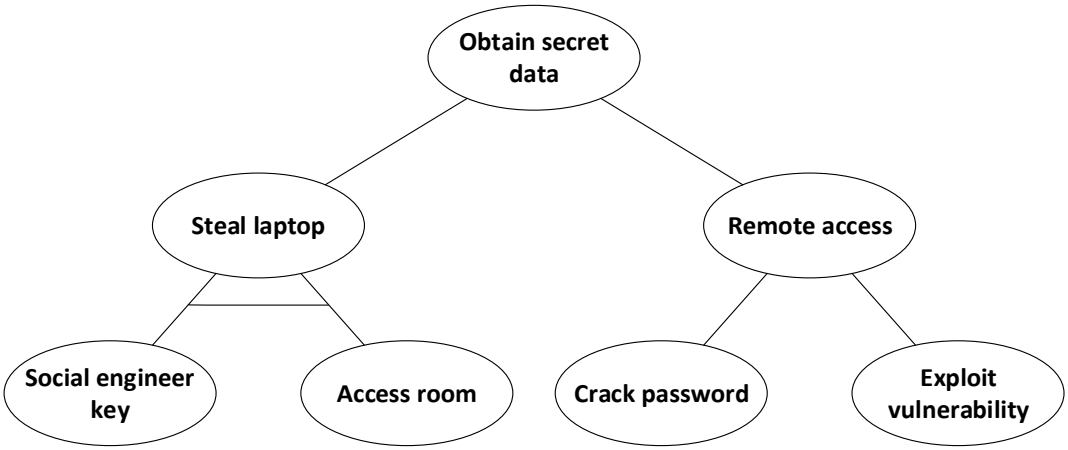


*Figure 4: Attack tree visualization of example (Based on Pieters et al., 2014)*

An example question that could be answered with this attack tree is the following: what attack suite describes the cheapest attack to obtain the secret data? For this a cost parameter has to be assigned to each of the leaf nodes. The values that are assigned for this cost parameter are merely dummy values to explain the methodology and should not be interpreted as the true values. The costs for social engineering a key has been set to €100, for accessing the room €50, for cracking the password €200 and for exploiting a vulnerability €300. How to calculate the value for the parameters for higher level nodes depends on the parameter. For the costs parameter an AND node assumes a costs value of the sum of its child nodes, because each of the attack components in the child nodes has to be performed in order for the AND node to be performed. An OR node assumes a value equal to the minimum value of its child nodes, because the attacker only needs to perform one of the attack components in the child nodes and will thus choose the cheapest one. This results in the attack tree shown in Figure 6. What can be seen from this attack tree is that the attack suite consisting of the attack components on the left branch is less costly than the attack suites consisting of one of the attack components of the right branch. Therefore it is more likely that an attacker will perform the attacks from the left branch when considering the costs.

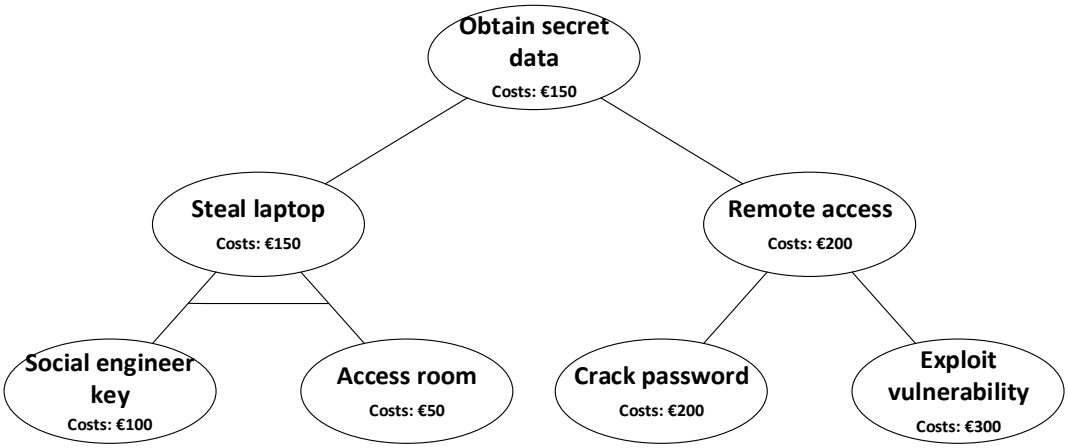


*Figure 6: Attack tree with costs parameter*

In order to illustrate how the method works for different types of parameters, another example is described. In this example the aim is to identify the attack suite that is most likely to result in a successful attack. In order to answer this question, the probability of success is the parameter of choice. A value for the parameter 'probability of success' has been assigned to each of the leaf nodes in the example attack tree. For the intermediate nodes and the root node the value of the probability of success parameter is calculated based on their child nodes. For an AND-node the value of the

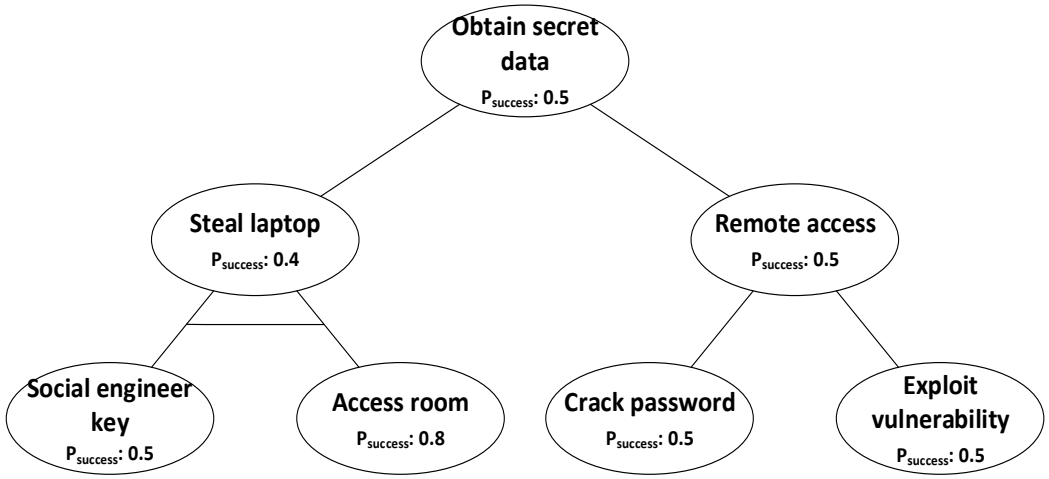


*Figure 5: Attack tree with probability of success parameter*

probability of success parameter is the product of the values of its child nodes, because all of the attack components of which it consists have to be a success for the AND-node to be a success. The value of the probability of success parameter for an OR-node is the maximum value of its child nodes, because only one of the alternative attacks has to be a success for the OR-node to be a success. In this case it is assumed that the attacker only attempts one of the child nodes in case of an OR node. The result of this can be found in Figure 5. From this attack tree it can be seen that the attack suites related to the right branch of the attack tree are more likely to result in a successful attack.

What is described here and what is illustrated by the two examples is the most basic use of attack trees. Over time, several extensions to the use of attack trees have been presented by literature. The extensions that were discussed in than just a single piece of literature are discussed in the following section.

### 3.1.3 Attack tree extensions

Three extensions of attack trees are discussed in this section. First the multi-parameter attack tree is described, which is used to take into account multiple interdependent parameters. The second extension is the serial model for attack trees, where the ordering of the attack components is taken into account. The last extension that is discussed is that of including countermeasures within the attack tree, which results in an attack-defence tree.

*Multi-parameter attack trees*

In the examples described in the previous section only one parameter is assigned to the nodes in the attack tree at a time. You may however also want to allocate multiple parameters that are interdependent to the nodes in the attack tree. In these cases you have to deal with multi-parameter attack trees. In their extensive literature review on directed acyclic graph-based attack and defence modelling, Kordy et al. (2014) indicate that the first notion on multi-parameter attack trees was made by Buldas et al. (2006). In their work they described a method to deal with multiple interdependent parameters in the attack tree methodology. The method is focussed on the expected outcome of an attack which is calculated based on the gains, costs and possible penalties of an attack (Buldas, Laud, Priisalu, Saarepera, & Willemson, 2006). During the analysis phase, with a rational attacker in mind, they look for the attack paths with the highest outcome. This method was later extended by Jürgenson & Willemson (2007) to also fit probabilistic interval estimations instead of exact values.

A problem with the method presented by Buldas et al. (2006) is that it is not consistent with the general framework presented by Mauw & Oostdijk (2006). In order to resolve this, Jürgenson & Willemson (2008) described some modifications to the method. Also a computational routine is given that calculates the maximum expected outcome for an attacker in a given attack tree (Jürgenson & Willemson, 2008). This computational routine was very inefficient, which lead to the improved routine presented in Jürgenson & Willemson (2010a).

An example is given here to explain how these multi-parameter attack trees work. The method described is based on the method of Jürgenson & Willemson (2008, 2010a). The attack tree used in the examples of the previous section is again used, but is now written as a Boolean formula. In order to do so, the four elementary attacks are indicated by $X_1$, $X_2$, $X_3$ and $X_4$, which correspond to the elementary attacks from left to right in the attack tree. The Boolean formula corresponding to the attack tree is then the following:

$$F = (X_1 \wedge X_2) \vee (X_3 \vee X_4)$$

Each of the attack components can either be attempted (set to True) or not (set to False). If *F* returns the value True, it means the root node is reached in the attack tree. The overall goal of the multi-parameter attack tree method is to optimize the outcome of the various attack suites. An attack suite is as described by Mauw & Oostdijk (2006) a combination of elementary attacks that an attacker is

assumed to attempt. Optimizing the outcome means looking for the maximum outcome for an attacker. The formula used for this optimization is:

$$Outcome = \max\{Outcome_\sigma : \sigma \subseteq X, F(\sigma := true) = true\}$$

An attack suite is represented by $\sigma$, where $\sigma$ consists of elements from the set *X*. The set *X* represents the complete set of elementary attacks. The outcome is calculated only for those attack suites that have the Boolean formula resulting in True, when the elementary attacks from within the attack suite are set to True and the other elementary attacks are set to False, which is in the formula indicated by $F(\sigma := true) = true\}$. The outcome of an attack suite is calculated based on the gains of reaching the root node and the expenses that go along with performing the attack components. The following formula is used:

$$Outcome_\sigma = p_\sigma \times Gains - \sum_{X_i \in \sigma} Expenses_i$$

The $p_\sigma$ stands for the probability of success of the attack suite $\sigma$. For each attack component $X_i$ from the attack suite, the expenses are summed up. The 'Gains' is a real number assigned to the attack tree to represent the gains obtained when reaching the goal in the root node.

The probability of success of an attack suite is determined by calculating the probability of success for the root node and the intermediate nodes, based on the probability of success of the elementary attacks. This results in a probability of success in the root node, which is the probability of success of the attack suite. The process of how this is done is the following:

- For all $X_i \notin \sigma$ the probability of success is set to 0
- For all $X_i \in \sigma$ the probability of success is left with the value it was assigned
- Now the probability of success of each non-leaf node *i* is calculated based on its child nodes *j*.
    - For an AND node the formula is: $\prod_{j=1}^{k} p_{i_j}$ (All need to be a success)
    - For an OR node the formula is: $1 - \prod_{j=1}^{k}(1 - p_{i_j})$ (1 minus the chance that all fail)

The expenses are built up from costs and penalties when being caught. The penalties when being caught for a successful attack are different from those of a failed attack.

- $\pi^+$ -> Expected penalty if attack was successful
- $\pi^-$-> Expected penalty if attack failed

The chance of being caught has already been taken into account within these parameters. The Expenses for each leaf node *i* are then represented by the formula:

$$Expenses_i = Cost_i + p_i \times \pi_i^+ + (1 - p_i) \times \pi_i^-$$

What can be taken from the mathematical system, is that the following parameters have to be assigned to each of the elementary attacks:

- Probability of success
- Cost
- Expected penalty if attack was successful
- Expected penalty if attack failed

Additionally a value should be assigned for the gains of reaching the overall goal in the root node. For this example the gains are considered to be 500 and the expenses are set equal to the cost for the sake of simplicity. Table 1 shows the parameter values for each of the elementary attacks. In Table 2

the attack suites that satisfy the root node have been listen with the associated outcome. The maximum outcome that is found is 50 and is reached by the attack suites {$X_1$,$X_2$} and {$X_3$}.

*Table 1: Elementary attacks with their assigned parameter values*

| Elementary attack | Probability of success | Expenses |
|---|---|---|
| $X_1$ | 0.5 | 100 |
| $X_2$ | 0.8 | 50 |
| $X_3$ | 0.5 | 200 |
| $X_4$ | 0.5 | 300 |

*Table 2: Satisfying attack suites and their calculated outcome*

| Attack suite (σ) | Probability of success | Expenses | Gains | Outcome |
|---|---|---|---|---|
| $X_1, X_2$ | 0.4 | 150 | 500 | 50 |
| $X_3$ | 0.5 | 200 | 500 | 50 |
| $X_4$ | 0.5 | 300 | 500 | -50 |
| $X_1, X_2, X_3$ | 0.7 | 350 | 500 | 0 |
| $X_1, X_2, X_4$ | 0.7 | 450 | 500 | -100 |
| $X_1, X_2, X_3, X_4$ | 0.85 | 650 | 500 | -225 |
| $X_1, X_3$ | 0.5 | 300 | 500 | -50 |
| $X_2, X_3$ | 0.5 | 250 | 500 | 0 |
| $X_1, X_4$ | 0.5 | 400 | 500 | -150 |
| $X_2, X_4$ | 0.5 | 350 | 500 | -100 |
| $X_3, X_4$ | 0.75 | 500 | 500 | -125 |
| $X_1, X_3, X_4$ | 0.75 | 600 | 500 | -225 |
| $X_2, X_3, X_4$ | 0.75 | 550 | 500 | -175 |

*Serial model for attack trees*

The use of attack trees that was described in the previous section can be seen as a parallel model, in which the attacker is considered to be indifferent about the order of the elementary attacks that he attempts to perform. It is however likely that an attacker would change strategy when one or more elementary attacks is not successfully performed. This is the reason for the introduction of the serial model by Jürgenson & Willemson (2010b) and Niitsoo (2010), where the elementary attacks are considered to be dependent on each other. First an order needs to be defined for all of the elementary attacks, which is the order in which the attacker will attempt each of the elementary attacks.

In the serial model of Jürgenson & Willemson an attacker will not perform an elementary attack if it does not have an effect on the overall probability of success of reaching the root node. This means that if a sub attack of an AND-node fails, the attacker will not attempt to perform any other sub attack of that AND-node. It also means that if a sub attack of an OR node succeeds, the attacker will not attempt to perform any other sub attack of that OR node (Jürgenson & Willemson, 2010b). In the serial model, there is thus a chance that an attacker does not attempt an attack from an attack suite. This may be because an elementary attack is redundant for reaching the root node. In this case the attacker would not make the expenses of the redundant elementary attack, which may lead to an higher expected outcome. How this is done mathematically is shown in the example later on.

Niitsoo attempted to improve the model by making the decision process more realistic. In order to reach this, he introduced the use of decision trees. With these decision trees the decision process of the attacker is modelled, which shows the order in which attacks or attempted and the outcome all of the attacks result in.

An example is given based on the serial model presented by Jürgenson & Willemson, 2010b), because this method does not need any additional graphs like the method presented by Niitsoo (2010). The same attack tree as in the previous examples is used, but letters have been assigned to the nodes for

easier reference. An elementary node is just like before represented by the letter *X* and an internal node is given the letter *Y*. Figure 7 presents this attack tree.

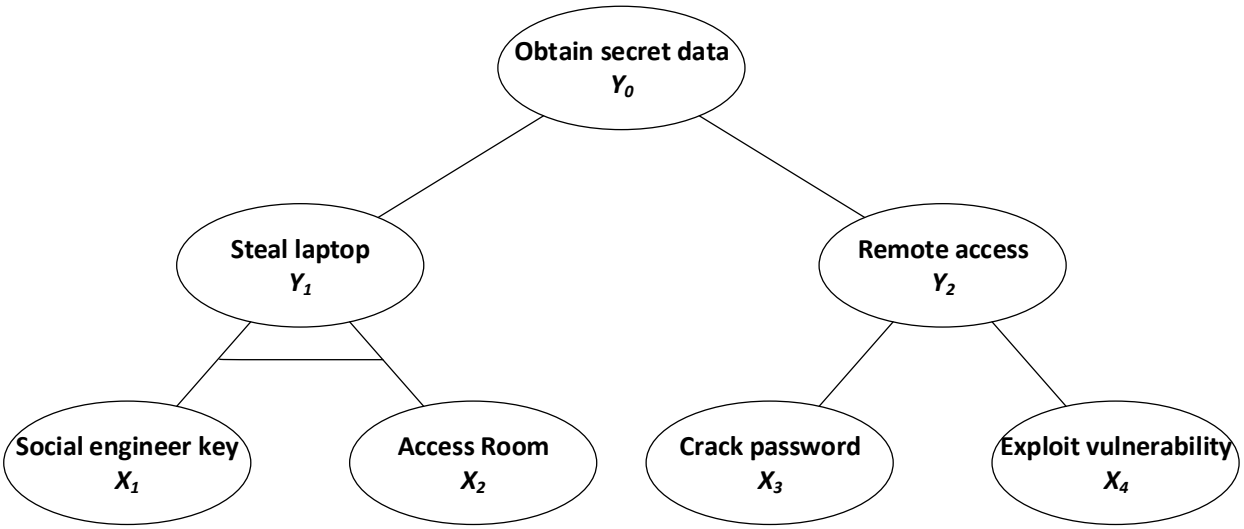


*Figure 7: Example attack tree for serial model (Based on Pieters et al., 2014)*

There are two main differences between the serial model and the parallel model, that are explained in this example. First there is an order in the elementary attacks that are in an attack suite. The other difference lies in the formula for the outcome. In the parallel model the following formula was used:

$$Outcome_\sigma = p_\sigma \times Gains - \sum_{X_i \in \sigma} Expenses_i$$

The serial model uses this formula:

$$Outcome_\alpha = p_\alpha \times Gains - \sum_{X_i \in \alpha} p_{\alpha,i} \times Expenses_i$$

In this formula $\alpha$ represents a certain permutation of the chosen attack suite. The probability in front of the expenses represents the chance that elementary attack *i* is performed considering the permutation $\alpha$. This probability is necessary, because it is possible that the attacker skips an attack if it does not have an effect on the overall probability of success of reaching the root node.

In order to show what effect the various orders of the attack suite can have, the example is worked out for two different orders. Also, because the probability that a certain elementary attack is performed is the main difference in the calculations, this is the only value that is calculated. The same probabilities of success for each of the elementary attacks is used. The two permutations that are worked out are {$X_1$, $X_2$, $X_3$, $X_4$} and {$X_3$, $X_4$, $X_2$, $X_1$}.

For calculating $p_{\alpha,i}$ three new parameters are introduced for each of the internal nodes and the root node, which are the probability of the node being proven true, the probability of the node being proven false and the probability that the node is yet undefined. These parameters are needed to calculate the chance that an attacker will attempt a certain elementary attack. These three probabilities always add up to a total of one. At every step of performing one of the elementary attacks, these three probabilities are again calculated. The value of the three probabilities is calculated based on the child nodes and the way to calculate it is dependent on the type of node. For an AND node *A* with child nodes *B* and *C* the following formulas are used:

$$A.t = B.t \times C.t$$
$$A.f = B.f + C.f - B.f \times C.f$$
$$A.u = 1 - A.t - A.f$$

The point *t*, point *f* and point *u* indicate the probability of proven true, false or undefined respectively. For an OR node these formulas are as follows:

$$A.t = B.t + C.t - B.t \times C.t$$
$$A.f = B.f \times C.f$$
$$A.u = 1 - A.t - A.f$$

For the first permutation {$X_1$, $X_2$, $X_3$, $X_4$} the calculations performed in each of the steps are presented in Table 3. Important to note here is that if an elementary attack is not yet performed its probability of proven true and its probability of proven false is set to zero. At the point in the permutation of attempting an elementary attack, its probability of proven true is set to the probability of success and its probability of proven false is set to one minus the probability of success.

*Table 3: Calculation of serial model for first permutation*

| Internal node | Probability of true | Probability of false | Probability of undefined |
|---|---|---|---|
| **First step ($X_1$)** | | | |
| $X_1$ | 0.5 | 0.5 | |
| $X_2$ | 0 | 0 | 1 |
| $X_3$ | 0 | 0 | 1 |
| $X_4$ | 0 | 0 | 1 |
| $Y_1$ | 0.5 * 0 = 0 | 0.5 + 0 – 0.5 * 0 = 0.5 | 0.5 |
| $Y_2$ | 0 + 0 – 0 * 0 = 0 | 0 * 0 = 0 | 1 |
| $Y_0$ | 0 + 0 – 0 * 0 = 0 | 0.5 * 0 = 0 | 1 |
| **Second step ($X_2$)** | | | |
| $X_1$ | 0.5 | 0.5 | 0 |
| $X_2$ | 0.8 | 0.2 | 0 |
| $X_3$ | 0 | 0 | 1 |
| $X_4$ | 0 | 0 | 1 |
| $Y_1$ | 0.5 * 0.8 = 0.4 | 0.5 + 0.2 – 0.5 * 0.2 = 0.6 | 0 |
| $Y_2$ | 0 + 0 – 0 * 0 = 0 | 0 * 0 = 0 | 1 |
| $Y_0$ | 0.4 + 0 – 0 * 0 = 0.4 | 0.6 * 0 = 0 | 0.6 |
| **Third step ($X_3$)** | | | |
| $X_1$ | 0.5 | 0.5 | 0 |
| $X_2$ | 0.8 | 0.2 | 0 |
| $X_3$ | 0.5 | 0.5 | 0 |
| $X_4$ | 0 | 0 | 1 |
| $Y_1$ | 0.5 * 0.8 = 0.4 | 0.5 + 0.2 – 0.5 * 0.2 = 0.6 | 0 |
| $Y_2$ | 0.5 + 0 – 0.5 * 0 = 0.5 | 0.5 * 0 = 0 | 0.5 |
| $Y_0$ | 0.4 + 0.5 – 0.4 * 0.5 = 0.7 | 0.6 * 0 = 0 | 0.3 |
| **Fourth step ($X_4$)** | | | |
| $X_1$ | 0.5 | 0.5 | 0 |
| $X_2$ | 0.8 | 0.2 | 0 |
| $X_3$ | 0.5 | 0.5 | 0 |
| $X_4$ | 0.5 | 0.5 | 0 |
| $Y_1$ | 0.5 * 0.8 = 0.4 | 0.5 + 0.2 – 0.5 * 0.2 = 0.6 | 0 |
| $Y_2$ | 0.5 + 0.5 – 0.5 * 0.5 = 0.75 | 0.5 * 0.5 = 0.25 | 0 |
| $Y_0$ | 0.4 + 0.75 – 0.4 * 0.75 = 0.85 | 0.6 * 0.25 = 0.15 | 0 |

With the help of these calculations the probability that an attack is attempted can be calculated. An elementary attack will only be performed if it has an influence on the overall probability of success, which is only the case if each of the internal nodes in its path to the root node and the root node itself are still undefined (Jürgenson & Willemson, 2010b). In Table 3 the probabilities of being undefined of the internal nodes and the root node are given. When calculating the probability of an attack being attempted or not, you always have to look at the probability of undefined values at the previous step of the permutation. For the first elementary attack in the permutation ($X_1$) the probability of attempting is 1, because the root node and all the internal nodes are undefined at the beginning of the attack. The second elementary attack ($X_2$) will only be attempted if the internal nodes on its path and the root node itself ($Y_1$ and $Y_0$) are still undefined after performing the previous elementary attack in the permutation. The probability of this happening is calculated by multiplying the probabilities of being undefined of these nodes after the first step. These calculations have been performed for each elementary attack and the results are shown in Table 4.

*Table 4: Probability of attempting an attack in the serial model for the first permutation*

| Leaf node | Probability of success | Probability of attempting the attack |
|---|---|---|
| $X_1$ | 0.5 | 1 |
| $X_2$ | 0.8 | 0.5 * 1 = 0.5 |
| $X_3$ | 0.5 | 1 * 0.6 |
| $X_4$ | 0.5 | 0.5 * 0.3 = 0.15 |

The same calculations have been performed for the second permutations {$X_3$, $X_4$, $X_2$, $X_1$} as well. The results of this are shown in Table 5 and Table 6. When comparing Table 4 and Table 6 it can be seen that the probability of an elementary attack being attempted depends on the permutation of the attack suite. For calculating the outcome of the attack suites these probabilities of attempting an attack are multiplied with the associated expenses. In this way the total expected expenses of an attack suite may be lower in the serial model than in the parallel model. The expected outcome may thus be higher in the serial model than in the parallel model.

*Table 5: Calculations of serial model for the second permutations*

| Internal node | Probability of proven true | Probability of proven false | Probability of undefined |
|---|---|---|---|
| **First step ($X_3$)** | | | |
| $X_1$ | 0 | 0 | 1 |
| $X_2$ | 0 | 0 | 1 |
| $X_3$ | 0.5 | 0.5 | 0 |
| $X_4$ | 0 | 0 | 1 |
| $Y_1$ | 0 * 0 = 0 | 0 + 0 – 0 * 0 = 0 | 1 |
| $Y_2$ | 0.5 + 0 – 0.5 * 0 = 0.5 | 0.5 * 0 = 0 | 0.5 |
| $Y_0$ | 0 + 0.5 – 0 * 0.5 = 0.5 | 0 * 0.5 = 0 | 0.5 |
| **Second step ($X_4$)** | | | |
| $X_1$ | 0 | 0 | 1 |
| $X_2$ | 0 | 0 | 1 |
| $X_3$ | 0.5 | 0.5 | 0 |
| $X_4$ | 0.5 | 0.5 | 0 |
| $Y_1$ | 0 * 0 = 0 | 0 + 0 – 0 * 0 = 0 | 1 |
| $Y_2$ | 0.5 + 0.5 – 0.5 * 0.5 = 0.75 | 0.5 * 0.5 = 0.25 | 0 |
| $Y_0$ | 0 + 0.75 – 0 * 0.75 = 0.75 | 0 * 0.25 = 0 | 0.25 |
| **Third step ($X_2$)** | | | |
| $X_1$ | 0 | 0 | 1 |

| | | | |
|---|---|---|---|
| $X_2$ | 0.8 | 0.2 | 0 |
| $X_3$ | 0.5 | 0.5 | 0 |
| $X_4$ | 0.5 | 0.5 | 0 |
| $Y_1$ | 0 * 0.8 = 0 | 0 + 0.2 – 0 * 0.2 = 0.2 | 0.8 |
| $Y_2$ | 0.5 + 0.5 – 0.5 * 0.5 = 0.75 | 0.5 * 0.5 = 0.25 | 0 |
| $Y_0$ | 0 + 0.75 – 0 * 0.75 = 0.75 | 0.2 * 0.25 = 0.125 | 0.125 |
| **Fourth step ($X_1$)** | | | |
| $X_1$ | 0.5 | 0.5 | 0 |
| $X_2$ | 0.8 | 0.2 | 0 |
| $X_3$ | 0.5 | 0.5 | 0 |
| $X_4$ | 0.5 | 0.5 | 0 |
| $Y_1$ | 0.5 * 0.8 = 0.4 | 0.5 + 0.2 – 0.5 * 0.2 = 0.6 | 0 |
| $Y_2$ | 0.5 + 0.5 – 0.5 * 0.5 = 0.75 | 0.5 * 0.5 = 0.25 | 0 |
| $Y_0$ | 0.4 + 0.75 – 0.4 * 0.75 = 0.85 | 0.6 * 0.25 = 0.15 | 0 |

*Table 6: Probability of attempting an attack in the serial model for the second permutation*

| Leaf node | Probability of success | Probability of attempting the attack |
|---|---|---|
| $X_1$ | 0.5 | 0.8 * 0.125 = 0.1 |
| $X_2$ | 0.8 | 1 * 0.25 = 0.25 |
| $X_3$ | 0.5 | 1 |
| $X_4$ | 0.5 | 0.5 * 0.5 = 0.25 |

*Attack-defence trees*

Another extension to the attack tree methodology is the inclusion of defence nodes to represent what countermeasures are in place. The foundations for this method are presented by Kordy et al. (2011) and an example is presented in Figure 8. In this attack-defence tree, defence nodes are shown as rectangles and are connected to the attack on which they have an influence by dotted lines. The attack-defence tree is intended to overcome the limitation of taking into account the countermeasures that are already in place. In the attack-defence tree methodology it is also possible to see how the analysis changes in the case that countermeasures are implemented. Just like the attack nodes, defence nodes can also be either AND nodes or OR nodes. Both attack nodes and defence nodes can have a child node of the other type indicating that the attack or countermeasure can be prevented or lowered in probability of success. Also the attack could result in extra costs for the attacker if s/he wants to overcome the countermeasure. An earlier notion of the use of defences in the attack tree methodology can be found in the paper by Edge et al. (2007), where next to the attack tree a protection tree is formed which corresponds to the attack tree and its metrics.

In order to show how the introduction of a countermeasure can influence the outcome of the attack tree, the example earlier used is extended. For both the costs and probability of success parameter earlier used, the influence of the introduction of a defence node is discussed.

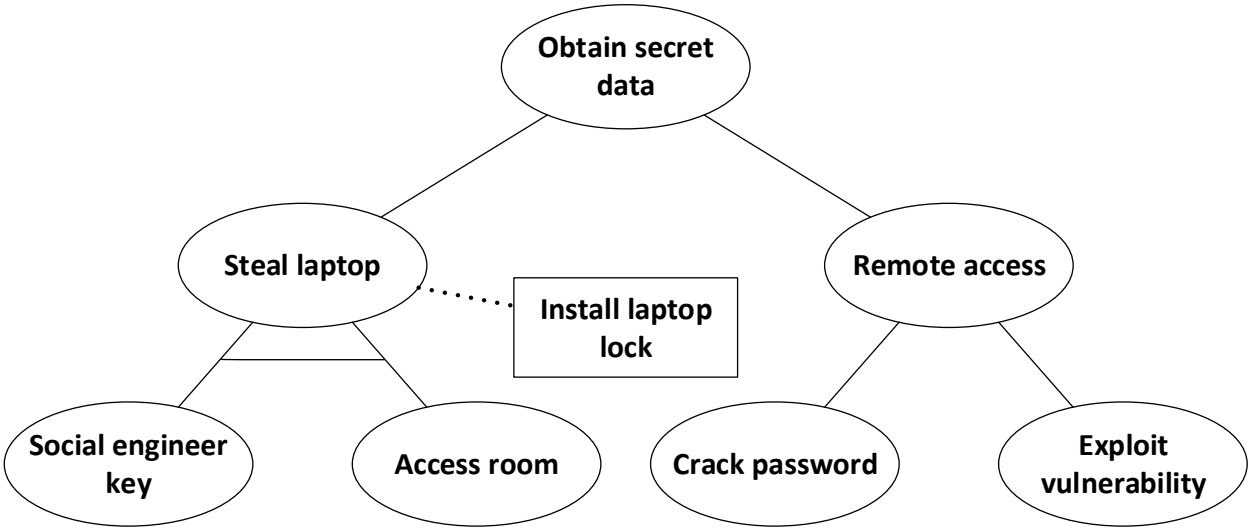


*Figure 8: Example of an attack-defence tree*

Figure 9 shows the attack-defence tree with the costs parameter. The costs that are shown in the defence node, are the costs for the defender to put the countermeasure in place. The attack node below it shows the costs for the attacker to overcome the countermeasure. This means that the steal laptop attack brings another €100 worth of costs with it, if the defender spends €100 on the laptop lock. The most likely path changed and is now via the right branch.

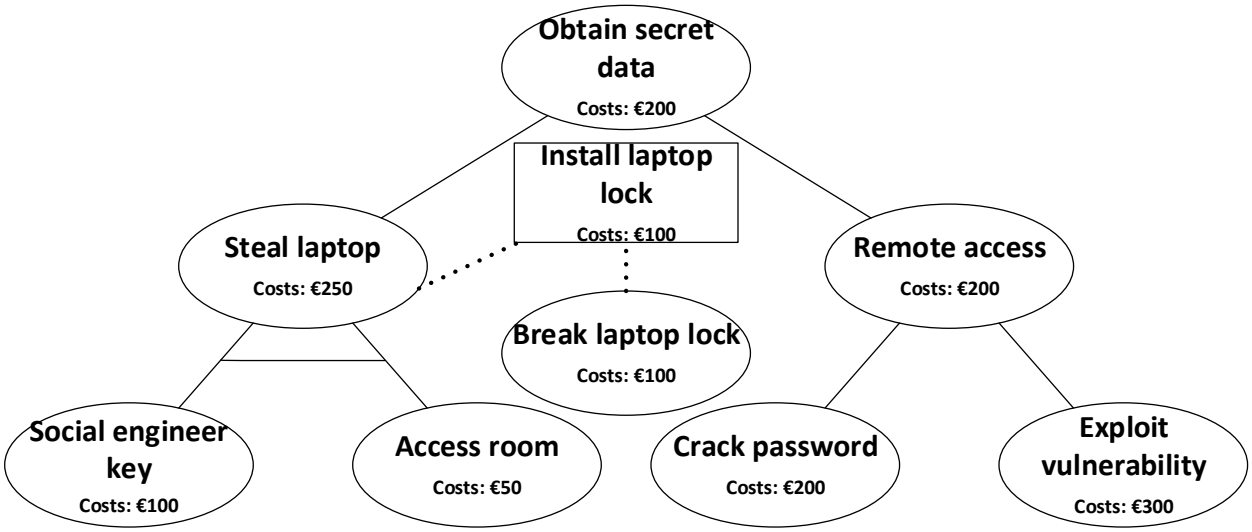


*Figure 9: Attack-defence tree with costs parameter*

The probability of success parameter in a defence node shows the probability for the successful implementation of the countermeasure. For this analysis an additional attack node to overcome the defence node is not necessary and is therefore not be included. Assuming the probability of success of a defence node is *x*, the probability of success of the attack node is multiplied by (1-*x*), which is the probability that the countermeasure will not be successfully put in place. This has been done for the example attack tree and the result is found in Figure 10. What results from this attack tree is that the attack in the right branch is the most likely one to be performed.

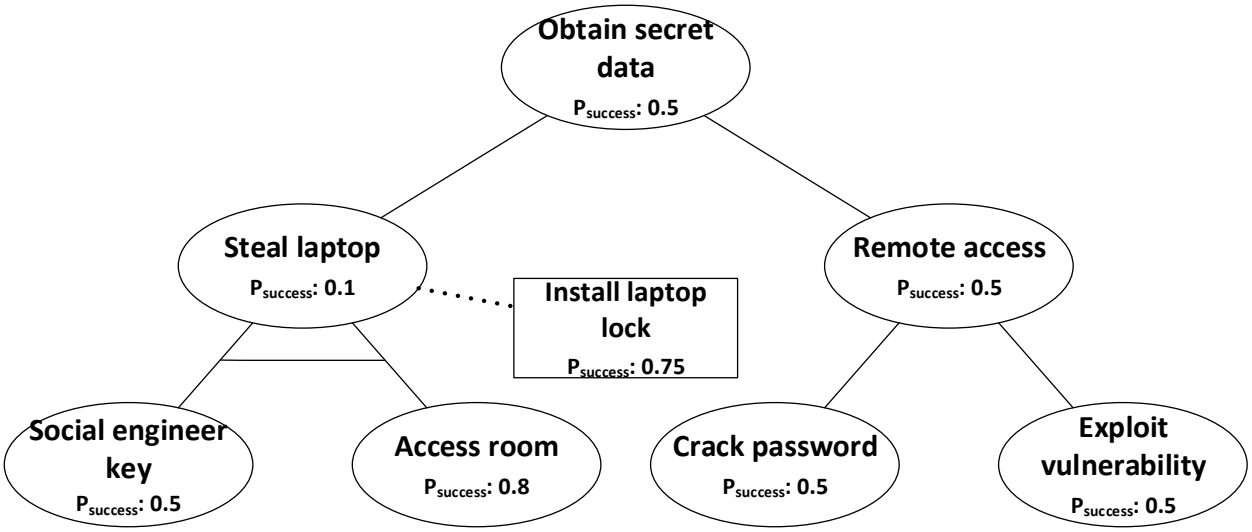


*Figure 10: Attack-defence tree with probability of success parameter*

## 3.2 Attacker profiles

With the use of attack trees, one tries to identify the attack suites that an attacker is most likely to use when trying to perform a certain attack. In order to do so, parameters are allocated to the attacks that have a possible influence on the attack suite that will be chosen by the attacker. The overall community of attackers is however not homogenous and the values of the parameter may vary for the various types of attackers. It is for example probable to think that the probability of success is higher for a very skilled attacker in comparison to an attacker who has low skills. In order to deal with this, Schneier already stated that he reasons from a certain type of attacker when allocating the values of the parameters to the attacks in the attack tree (1999). Also Weiss does not neglect the fact that there are various types of attackers, but states that in his method it is not necessary to assume the different types of attackers, because the method deals with the worst case scenario. For security experts it is however important to deal with the most realistic scenario, because this will help to correctly allocate the resources available for putting in place countermeasures. In order to get insight in this realistic scenario it is interesting to be able to test for various types of attackers. This way the security experts will be able to analyse all possible scenarios. This could be done using Schneier's method, where for each type of attacker the parameters are given new values. A more efficient way however, would be by making the parameters independent of the attacker.

By considering the characteristics of the attacker separately, the parameters of the attacks in the attack tree can be made attacker independent. The question is then, what characteristics to consider. Because you are trying to predict what attack suites are more likely to be used by an attacker, those characteristics that influence the decision on what attack suite to use are interesting. In literature these different attacker characteristics have been used to form attacker profiles. Each of these profiles describes a certain type of attacker with the associated characteristics. Section 3.2.1 describes what different attacker profiles the literature presents and also how these profiles have been formed through time. The report focusses on the motivation of the attacker. The different types of motivation are therefore clearly listed in 3.2.2.

### 3.2.1 Attacker profiles

Some time ago it was generally assumed that the community of attackers was a homogenous group of which the members were profiled as "pimply-faced 14-year-old kids (mostly male) with anti-social tendencies and an addiction to Sci-Fi" (Barber, 2001, p14). Long before that however, there were already some researchers that were trying to define this community of attackers in more detail. In these attempts to define the community it was understood that multiple types of attackers exist.
Smith & Rupp (2002) discuss some of the earlier research done to define the hacker community. The studies that they describe categorize the attackers on the basis of their technological skill level and on

their motivation for performing the attacks. The categories that were formed in these early studies were however inconsistent. Some presented the least skilled group as being curious and others as being motivated by economic gains. Also opinions differ on the motivation of the most skilled group. One study states that the most skilled group is motivated by curiosity, the other stated that vandalism and economic benefits are the drivers (Smith & Rupp, 2002).

Somewhat later, Barber (2001) described in some more depth three main groups of attackers that exist within the attacker community, being script-kiddies, hackers and crackers. The group of script-kiddies actually comes closest to the general opinion on attackers. They are usually male kids that are still in a very amateur stage of hacking, who mostly use tools that are provided to them by more professional attackers. Which immediately leads to the second group, hackers, which consists of more experienced people writing their own tools to perform their actions. The last group, crackers, are like hackers but differ in the motivation behind their actions. Hackers are considered to just be curious of what they can achieve, but crackers are actually causing damage to people or companies on purpose (Barber, 2001).

Rogers (2006) identified more types of attackers. He identified eight different types of attacker profiles which were formed based on the technical abilities and the motivation of the attacker. The four types of motivation that he identified are revenge, financial, curiosity and notoriety. Using the skill level and the motivation of the different attacker profiles, he created an overview which is presented in Figure 11. The graph is structured in the way that the more skilled a group of attackers is, the further away it is from the centre. The four quadrants represent the four categories of motivation in which the attacker profiles are put (Rogers, 2006). The group Novice (NV) for example is considered to consist of attackers that have a low skill level and are mainly motivated by curiosity. The following abbreviations are used in the overview:

- Novice (NV)
- Cyber-punks (CP)
- Internals (IN)
- Petty Thieves (PT)
- Virus Writers (VW)
- Old Guard hackers (OG)
- Professional Criminals (PC)
- Information Warriors (IW)

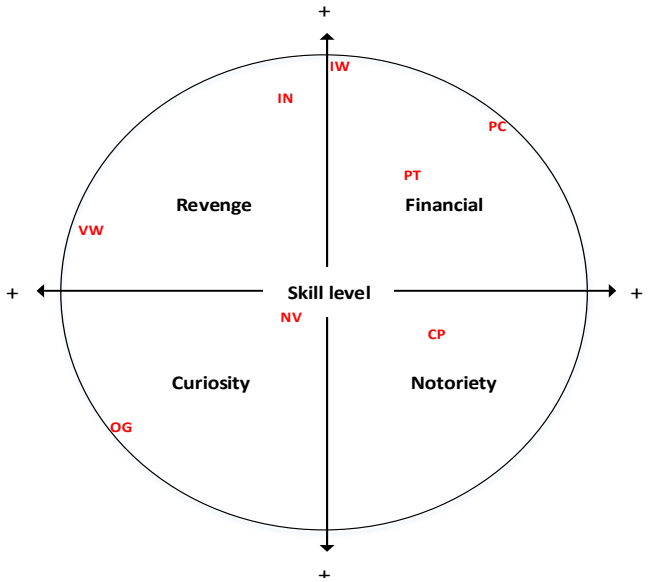


*Figure 11: Attacker profiles categorization based on skill and motivation (Based on Rogers, 2006)*

Rogers already indicates that the eight groups he identifies might not be enough to fully indicate all the different types of attackers. This seems to be true when you look at the Threat Agent Library (TAL)

set up by Intel. This TAL is used by security experts to identify possible threat agents and to understand the threats they bring along. In this TAL 21 different profiles have been formed on the basis of eight attributes (Casey, 2007). The Threat Agent Library is visualized in appendix A. What should be noted however is that these 21 profiles are not all actual 'attacker profiles' because some threat agents are also notified as non-hostile. These non-hostile threat agents are for example employees that accidentally open a phishing mail which is actually also a threat for the cybersecurity. The attributes that the TAL is built on are intent, access, outcome, limits, resource, skill level, objective, visibility. This TAL was later expanded by adding the attribute motivation (Casey, 2015).

All of the studies analysed, discuss the motivation of the attacker as an important aspect of the attacker profile. What types of motivations attackers may have is however not defined in the same way in each of the studies. The next section therefore forms an overview that clearly defines the various types of motivation that were found in literature.

### 3.2.2 Motivation of attackers

As was seen in the previous sections, motivation is an important part of attacker profiles. There are however two ways in which this motivation can be interpreted. The motivation can say something about the reason why an attacker is performing an attack or the motivation can say something about how high the level of interest of the attacker is for performing the attack. Casey (2015) describes these two forms of motivation as cause and drive respectively. The cause describes why an attacker wants to perform an attack and the drive describes how badly the attacker wants do perform an attack. This research focusses on why the attacker performs an attack and thus focusses on the cause. From now on these different causes will be referred to as different types of motivation. In this section an overview is presented of these types of motivation an attacker may have.

The different types of motivation of an attacker are categorized in two different categories. The first category of motivations is extrinsic, where the outcome of the behaviour, or in this case attack, is the thing that motivates the attacker. The second category is intrinsic, where the motivation comes from performing the attack itself (Lindenberg, 2001). The intrinsic motivations can also be seen as psychological motivations (Kshetri, 2006). This section first discusses the extrinsic motivations, followed by the intrinsic motivation.

#### *Extrinsic motivations*

When discussing the extrinsic motivations of attackers, it is important to see what an attack could possibly result in, that can be interesting for an attacker. The three types of motivation that were found and are discussed are financial benefits, causing damage and knowledge gaining.

The first and probably most obvious extrinsic motivation for an attacker is the financial benefit the attack can result in. The financial benefits are mentioned as a motivation of attackers by multiple studies (Turgeman-Goldschmidt, 2005; Kshetir, 2006; Thycotic Software Ltd, 2014; Casey, 2015;). Performing cyberattacks is often seen by attackers as a good way to earn money (Blau, 2004). An attacker could for example commit fraud or extort other people with the help of cyberattacks. Cybercrime in general has a whole economy around it, based on so called zero-day exploits, which is a leak within a system for which no patch is available yet (Greenberg, 2015). When an attacker finds such a zero-day exploit s/he could choose to report it to the company that made the software, share it with the Zero Day Initiative or he can sell it to anyone who is interested (Greenberg, 2012). Recently a new marketplace has opened on which zero-day exploits can be sold (Greenberg, 2015), which underlines this economy around cybercrime. Attack trees however, focus on the defence of one certain system, where the selling of zero-days will not be the motivation. This phenomenon does however proof that money is an important factor when it comes to cybercrime. The financial motivation is also mentioned by Smith & Rupp (2002), Barber (2001) and Rogers (2006).

Causing damage is another type of extrinsic motivation that an attacker may have. The willingness to cause damage can come from various reasons. Two possible reasons can be distilled

from the motivations presented by Casey (2015), which are disgruntlement and ideology. In these two cases the attacker is considered to be willing to cause harm based on revenge or based on certain ideas s/he has. Casey (2015) mentions various sources of idealism from which the ideas may come, which are an attacker's sense of morality, justice or political loyalty. An attacker's morality is also mentioned in the survey by Thycotic Software Ltd (2014) as an important motivation for attackers. Revenge as a motivation is also mentioned by Rogers (2006) and Turgeman-Goldschmidt (2005).In this research, when an attacker is motivated by revenge or ideology, s/he is considered to be motivated to cause damage to the target.

The last type of extrinsic motivation is the knowledge that an attacker can gain from performing an attack. Literature uses various terms for what in this research is defined as the motivation knowledge gaining. Casey (2015) talks about organizational gain which is defined as gaining advantage over a competitor. The way in which advantage is reached is through the theft of information which in this research is defined as knowledge. Another motivation listed by Casey that can partly be put within this type of motivation is dominance, which is defined as the attempt to get superiority over someone else. This can also be reached by stealing information (Casey, 2015). Turgeman-Goldschmidt (2005) describes the motivation to gain knowledge as voyeurism, which also entails the motivation to gain insight into the confidential and into secrets.

*Intrinsic motivations*

The intrinsic motivations can be found by figuring out in what way performing the attack itself can possibly be beneficial for the attacker. The attacker will gain something from performing the attack in a psychological sense in this case. Two types of intrinsic motivation are found and described, which are pleasure and notoriety.

In multiple studies it was concluded that a very important reason for attackers to perform cyberattacks is the pleasure it brings them. Attackers motivated by this pleasure seeking motivation are performing attacks for fun. Multiple reasons are given in literature for where this fun or pleasure comes from. Casey (2015) combined these reasons in the personal satisfaction motivation, under which he names curiosity and thrill seeking. Also boredom or the willingness to test one's ability can lead an attacker to perform attacks to seek pleasure (Thycotic Software Ltd, 2014). Turgeman-Goldschmidt (2005) separates the motivations pleasure seeking and curiosity as they are separately mentioned by the author's interviewees. In this research it is assumed that someone that is performing an attack from curiosity is mainly gaining pleasure from performing the attack. Curiosity will therefore also be gathered under the pleasure seeking motivation. Curiosity is also mentioned by Smith & Rupp (2002), Barber (2001) and Rogers (2006) as a motivation for attackers.

Another psychological incentive for attackers lies within the community of attackers that exist. Attackers may be motivated by gaining notoriety within such a community, which brings them psychological benefits. Performing certain attacks may lead to a gain of respect from the peers of the attacker (Kshetri, 2006). Also Casey (2015) lists the notoriety motivation and states that an attacker with this motivation aims to become well known for performing attacks. Turgeman-Goldschmidt (2005) uses the term computer virtuosity, which also included attackers that perform attacks to gain respect from their peers. From the survey performed by Thycotic Software Ltd (2014) it was also concluded that notoriety is a possible motivation for attackers. Rogers (2006) mentions notoriety as a motivation for attackers as well.

*Reflection*

Two notes are added to the analysis of the different types of motivation presented above. First, the different types of motivations are discussed in a random order and no conclusions can be drawn from the list about the number of hackers that have that type of motivation. The reason for not including such an order of 'more important' motivations, lies in the fact that there seems to be no consensus on the importance of the various types of motivation. Some studies conclude that the psychological motivations are dominant (Henych, 2001; Turgeman-Goldschmidt, 2005; Thycotic Software Ltd, 2014)

and others conclude that economic motives are becoming more and more important (Blau, 2004; Kshetri, 2006).

Another note is made on the motivations that are listed by Casey (2015). In his analysis he identifies three other types of motivations that are not included in the list above. Two of those are accidental and coercion of which the former means that the attacker harms the system accidentally and the latter means that the attacker is forced by someone else to perform the attack. These types of motivation are not included in this research, because these belong to non-hostile attackers. The third motivation that is not included in this research is the unpredictable motivation. An attacker with this motivation is considered to act "without identifiable reason or purpose" (Casey, 2015 p7). As the name of the motivation already suggests, attacks by this type of attacker are unpredictable. Because the attack tree methodology is used for predicting the attacks that will be performed, this type of motivation is not suitable to include in this research.

## 3.3 Attack trees and attacker profiles combined

So far an in depth description of attack trees and attacker profiles has been given. In this research, the two concepts are combined. Some literature is available on the combination of the two, of which the most important findings are discussed here. Two known methods of using attacker profiles within the attack tree methodology are described. The first one is presented by Grunske & Joyce (2008) and is described in section 3.3.1. The other by Lenin, Willemson & Sari (2014), which is described in section 3.3.2. Section 3.3.3 provides a reflection on the differences between the two methods.

### 3.3.1 Attacker profiles to determine the probability of occurrence of an attack

The method of Grunske & Joyce is based on calculating the probability of occurrence of an attack. This probability of occurrence is calculated with the help of so called attack profiles. These attack profiles consist of the attacker characteristics and of some characteristics of the environment in which the system operates. In these attack profiles they combine the system properties and the attacker properties. The first step in their method is to prune the original attack tree based on the environmental conditions and some of the attacker characteristics (Grunske & Joyce, 2008). The pruning of the attack tree is actually scaling the attack tree down, by removing the attacks that are considered impossible. In the method of Grunske & Joyce, attacks can be considered impossible in three ways:

- The required environmental conditions are not met by the attack profile
- The required capabilities of the attacker are not met by the attack profile
- The required resources for the attacker are not met by the attack profile

After pruning the attack tree, the probability of occurrence of an attack is calculated based on two parameters. These parameters are 'attacker motivation and ranks', which is a parameter from the attack profile, and 'attacker risk and cost', which is a parameter from the attack itself. The 'attacker motivation and ranks' parameter has its value based on the amount of attackers that are present and on how motivated they are, or the drive they have, to perform an attack. The 'attacker risk and cost' parameter is based on the probability of the attacker being caught and the penalties that are associated with being caught. Several possible inputs for calculating these two metrics are given, but the authors leave the exact definition of these metrics and the function in which they are used for calculating the probability of occurrence for future work. Because this method is not fully worked out, no example is given.

### 3.3.2 Attacker profiles to determine possibility of an attack

Lenin et al. (2014) explain that attacks are both dependent on the system properties, as well as the attacker properties. Attacker profiles are used in the attack tree methodology to separate these two types of properties. The system properties are represented in the attack tree parameters and the attacker properties are represented in the attacker profiles. The usefulness of this is underlined by

Pieters et al. (2014), who state that the parameter values in the attack tree are made independent of the type of attacker by using attacker profiles. This independency of the attack tree parameters adds to the flexibility of the quantitative security analysis, because the attack tree can be analysed for different types of attackers, without having to update the parameter values (Lenin et al., 2014).

In their method, Lenin et al. (2014) use three types of system properties and three types of attacker properties. The system properties are the expenses, the difficulty and the minimal required attack time. The attacker properties are the budget, the skill and the available time. In their calculations, they use these parameters to determine whether it is considered possible for an attacker to perform a certain attack suite. This is done by using the parameters on a constraint basis. The attacker should have enough budget to meet the costs, enough skill to meet the difficulty and enough time to meet the minimal required time. By doing so, only the attacker profile satisfying attack suites are left for the analysis.

To give an example of how this works, the attack tree from the earlier example to explain the multi-parameter attack trees is used, which is described by the following Boolean Formula:

$$F = (X_1 \wedge X_2) \vee (X_3 \vee X_4)$$

For this example only the budget of the attacker and the expenses of the attack are taken into account. Table 7 shows the expenses associated with each of the elementary attacks. With the help of these values, the expenses for the satisfying attack suites can be calculated. If we now assume an attacker with a budget of 400, some attack suites become impossible to perform for that attacker. Table 8 shows whether the attack suites are profile satisfying or not. In the same way attack suites can be cut off by checking whether there are elementary attacks in the attack suite have a higher difficulty level than the skill level of the attacker and by checking whether there are elementary attacks in the attack suite that have a higher minimal attack time than the available time of the attacker.

*Table 7: Elementary attacks and their associated expenses*

| Elementary attack | Expenses |
|---|---|
| $X_1$ | 100 |
| $X_2$ | 50 |
| $X_3$ | 200 |
| $X_4$ | 300 |

*Table 8: Attack suites with their associated expenses and indication of being profile satisfying*

| Attack suite (σ) | Expenses | Profile satisfying? |
|---|---|---|
| $X_1, X_2$ | 150 | Yes |
| $X_3$ | 200 | Yes |
| $X_4$ | 300 | Yes |
| $X_1, X_2, X_3$ | 350 | Yes |
| $X_1, X_2, X_4$ | 450 | No |
| $X_1, X_2, X_3, X_4$ | 650 | No |
| $X_1, X_3$ | 300 | Yes |
| $X_2, X_3$ | 250 | Yes |
| $X_1, X_4$ | 400 | Yes |
| $X_2, X_4$ | 350 | Yes |
| $X_3, X_4$ | 500 | No |
| $X_1, X_3, X_4$ | 600 | No |
| $X_2, X_3, X_4$ | 550 | No |

Lenin et al. (2014) state that this constraint basis, is only one possible way to use the attacker profiles. Pieters et al. (2014) describe another way to incorporate the difficulty of the attack and the skill of the attacker. The probability of success is considered to be dependent on the difficulty and skill parameter. If an attack is more difficult, the probability of success will be lower assuming the skill of the attacker stays the same. Likewise, when the skill of an attacker is higher, the probability of success will be higher. The function for the probability of success is chosen in such a way that the probability of success is 0.5 when the difficulty and the skill are equal.

To illustrate this method with an example, the attack tree with the probability of success parameter assigned is used, which is shown in Figure 5 on p11. In the new case a difficulty for each attack is assigned to the branch and the probability of success that is shown in the attack nodes is calculated with the following formula $P_{success} = \left(e^{\beta-\delta}\right)/(1+e^{\beta-\delta})$, in which beta represents the skill of the attacker and delta represents the difficulty of the attack (Pieters et al. 2014). Figure 12 shows the resulting attack tree with an attacker with a skill of 1 assumed. As in the example presented before, the attacker is here assumed to attempt only one of the attacks in the child nodes of an OR node.

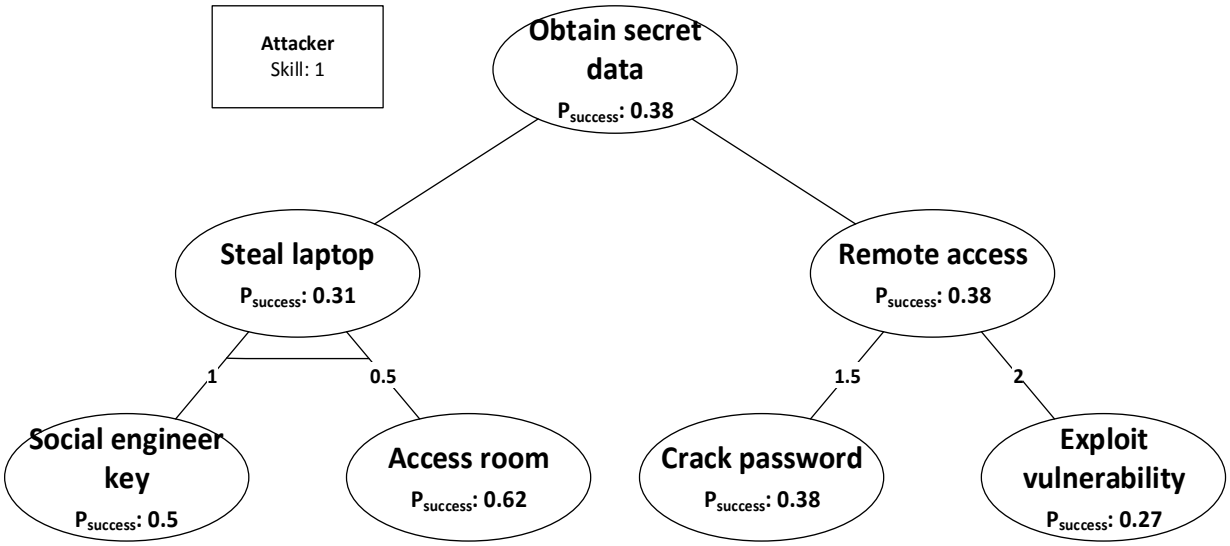


*Figure 12: Attack tree with probability of success parameter based on skill and difficulty*

### 3.3.3 Comparison of the methods

The way in which both of the methods described use the attacker profiles in its basic form appears to be the same. Both Grunske & Joyce (2008) and Lenin et al. (2014) use attacker characteristics to cut of attacks from the tree that the attacker is assumed unable to perform. The characteristics they use for this are the resources of the attacker and the attacker's skill. Lenin et al. (2014) splits these resources in the budget and the available time of the attacker. Lenin et al. (2014) do however mention that there are other uses possible with their method as well. This was illustrated with the example presented in Figure 12.

Another big difference between the methods is what they try to predict with the attack tree. In the case of Grunske & Joyce (2008), the method is used to determine the probability of occurrence of an attack. Lenin et al. (2014) build forward on the method developed by Jürgenson & Willemson (2008; 2010a), where the goal is to determine the expected outcome of various attacks. The method by Grunsky & Joyce (2008) is also less worked out than that of Lenin et al. (2014), which makes it less useful at this point.

## 3.4 Knowledge gap

In this chapter the state of the art in the field of attack trees and attacker profiles has been explored. The use of attack trees has over time become more and more sophisticated in an attempt to predict the behaviour of an attacker. Attacker characteristics that have an influence of this behaviour are taken into account while analysing the attack tree. In order to be able to reuse the attack tree for different types of attackers without having to update the parameters, it is attempted to make the parameters in the attack tree attacker independent. So far methods have been described that use the attacker's resources and the attacker's skill in the attack tree analysis. From the descriptions of attacker profiles in literature it was concluded that the motivation of attackers is also an important characteristic. No methods have however been found in literature that include the various types of motivation an attacker may have in the attack tree analysis. This is the gap that this research attempts to resolve. The remainder of this report describes the steps that have been gone through to come up with a design for an extended attack tree methodology that takes into account the motivation of attackers in the analysis.

# 4 Potential added value

From the previous chapter it was concluded that the motivation of attackers has not yet been taken into account in the attack tree methodology. It is not yet known how the motivation of attackers can be included in the attack tree methodology. This chapter focusses on the value that including this motivation of attackers can add to the overall methodology. The chapter formulates an answer to the second sub question; *What value could the inclusion of motivation in the use of attack trees add to the information gained from the attack tree analysis?*.

The answer to the question is found by analysing gaps in the current method as presented by Lenin et al. (2014) that can possibly be solved by including the motivation of attackers. The reason for using this method instead of the one presented by Grunske & Joyce (2008) lies in the fact that the latter is not fully worked out. Also within the context of this research, the method developed by researchers of Cybernetica is preferred. In order to identify the possible added value of including the motivation of the attacker, first the parameters on which the motivation has an influence need to be determined. These are the parameters that need to be made independent of the attacker to make the attack tree reusable for various types of attackers. After identifying these parameters, it is can be determined how changing these parameters can add value to the attack tree methodology. In section 4.1 the parameters are identified and in section 4.2 the way in which these parameters need to be altered is discussed. The added value is summarized in section 4.3.

## 4.1 Added value to the parameters

As described in the previous chapter, in the attack tree methodology parameters are assigned to each of the elementary attacks in the attack tree to represent the properties of the attack. In the approach presented by Lenin et al. (2014) some of these parameters were made independent of the type of attacker by separating some attacker characteristics from these attack properties. The attacker characteristics that are used by Lenin et al. (2014) are budget, skill and time available. The attack properties used are expenses, difficulty, time needed, probability of success and gains.

As was discussed earlier, motivation is also an important characteristic of an attacker. The method of Lenin et al. (2014) has however not made the parameters independent of this attacker characteristic. This is the gap that the current research is trying to resolve by designing a framework that includes the motivation of attackers in the attack tree methodology. In order to make the parameters independent of this attacker motivation, it is first necessary to identify on which parameter(s) this attacker characteristic has an influence.

First of all, the expenses parameter is not considered to be influenced by the motivation of the attacker. An attack will for example not be more or less costly for an attacker motivated by gaining knowledge than by an attacker interested in notoriety within a community. No added value is to be reached with this parameter when including the motivation of attackers. Also the difficulty of the attack will not change for a differently motivated attacker.

The time needed for performing an attack will not change when an attacker is differently motivated. It is however possible to think that an attacker that is stronger motivated is able to perform the attack in less time than an attacker that is not very motivated. This is however the motivation in terms of drive that has a possible influence on the required attack time, which is thus not taken into account in this research.

Just like the required attack time parameter, the probability of success parameter would be influenced by the motivation of the attacker if the drive of the attacker would be considered. A stronger motivated attacker will probably do more effort and might keep trying, which could increase the probability of success. However, this report deals with the cause aspect of the motivation and not the drive aspect. In the light of this research, the motivation of attackers does not have an influence on the probability of success parameter.

The last attack property is the gains parameter. This gains parameter describes the pay-off that the attacker receives from performing the attack. This pay-off can depend on the motivation of the attacker, which is explained by an example. If an attacker is motivated by possible notoriety within a community performs an attack that will only result in a monetary gain and no notoriety at all, the pay-off for this attacker will be very low. If the same attack is performed by an attacker that is financially motivated, the pay-off will be higher. What can be seen from this is that the pay-off an attack results in, is dependent on the motivation of the attacker.

The most important parameter that the motivation of the attacker seems to have an influence on, is thus the gains parameter. Changing this parameter can possibly add great value, because the gains parameter is currently not very well defined in the attack tree methodology. In the method presented by Buldas et al. (2006), but also in the extensions of this method presented by Jürgenson & Willemson (2008; 2010a; 2010b) and Lenin et al. (2014), the gains is a global parameter, which means that whichever attack suite the attacker uses, the gains are always the same and the pay-offs are the same for every type of attacker.

To give an overview of the dependencies that exist between the attacker properties and the attack properties, Figure 13 has been formed. Only the attack and attacker properties used in the method of Lenin et al. (2014) are included with the addition of the motivation of the attacker. There are more properties that could be taken into account, like for example the available tools for an attacker. However, the focus of the current research is to include the motivation, which is why other properties not included in the method of Lenin et al. (2014) are not used in the diagram.

Some dependencies between the attack and the attacker properties are already taken into account in the method of Lenin et al. (2014). These dependencies are indicated by the green arrows. The orange arrow indicates the dependency that is the focus point of the current research that followed from the analysis described above. The red arrows are dependencies that also exist that have not yet been taken into account by Lenin et al. (2014) and are also not taken into account in this research. Future research could focus on these dependencies. The dependency between the skill and the probability of success was already indicated by Lenin et al. (2014). The link between the drive of an attacker and the probability of success and the link between the probability of success and the required attack time were discussed above. The last arrow that is added is the dependency between the skill of the attacker and the required attack time. An attacker that is more skilled could require less time to perform an attack than an attacker that has a lower skill.

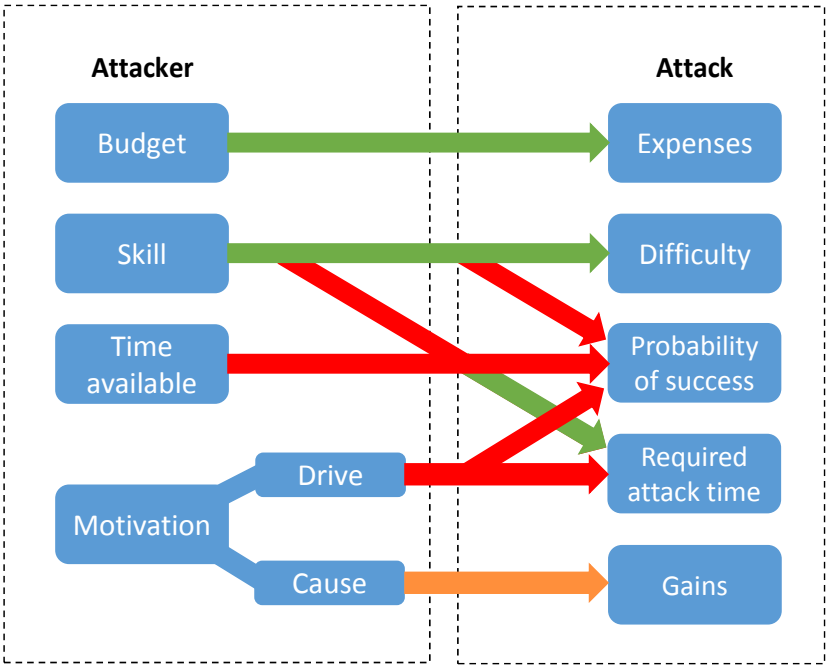


*Figure 13: Dependencies between attacker and attack properties*

## 4.2 Changing the gains parameter

In the previous section it was concluded that the pay-off of an attack depends on the type of attacker. In the current methodology the pay-off is equal to the value of the gains. In order to allow for the reuse of the attack tree, the gains parameter should be made independent of the attacker and the motivation of the attacker should be considered separately. In this way the pay-off can be determined for various types of attackers.

In the process of changing the gains parameter, it is also possible to make it more realistic. In the current methodology, The gains are assumed the same for every way in which the attacker can reach the root node of the attack tree. There are however conceivable situations in which the path of the attack should also have an influence on the gains. For example, stealing a laptop to obtain secret data results in more gains then obtaining the data via remote access, because the laptop itself is also worth something. By changing the gains parameter to be dependent on the attack path taken, it can be made more realistic.

A more realistic gains parameter is directly linked to a more realistic calculated outcome for the attacker. A more realistic calculated outcome, means a more realistic overview of the security of the system for the IT security expert.

## 4.3 Prospected added value

The aim of this chapter is to formulate possible added value of including the motivation of attackers in the attack tree methodology. It can be concluded that changing the gains parameter is necessary for including the motivation of attackers and in changing this parameter, the most value can be added. The following points summarize the prospected added value:

- The gains parameter is made independent of the type of attacker
- Various pay-offs are possible for variously motivated attackers
- The gains parameter is made more realistic

# 5 Forming the list of requirements

With the value that can be added in mind, this chapter describes a list requirements for the framework design of including the motivation of attackers in the attack tree methodology. This chapter thus gives an answer to the third sub question: *What are the requirements for a framework that includes the motivation of attackers in the use of attack trees?*

The list of requirements that is described, is build up from two different types of requirements. The first part consists of the constraints to which the design must conform. The second part is a list of dilemmas for which a choice is made during the design and development step. First the constraints are described in 5.1. The dilemmas are described in 5.2.

## 5.1 Constraints

From analysing the current methods and with the objective of the research in mind several constraints have been formed to which the framework to be designed has to comply. A total of four constraints have been formed and for each it is argued why it is a constraint. The constraints are formulated as should-sentences.

### C1 The framework should use multi-parameter attack trees

As was discussed before in the previous chapter, this research builds on the method presented by Lenin et al. (2014). This method uses multi-parameter attack trees, which means that the framework to be designed in this research also needs to use multi-parameter attack trees. Another reason for using multi-parameter attack trees lies in the fact that the main parameter to be adjusted in the new framework is the gains parameter. Only considering the gains would not lead to a useful analysis of the attack tree.

### C2 The framework should be able to deal with the various types of attacker motivation

The objective of this research is to make it possible to analyse attack trees for variously motivated attackers. In section 3.2.2 the types of motivation that an attacker may have were analysed. The following types of motivation were found:

- Financial benefits
- Causing damage
- Knowledge gaining
- Pleasure
- Notoriety

In order to reach the objective of the research, the framework should thus be able deal with each of these types of motivation. What should be noted is that an attacker does not necessarily have to be motivated by just one type of motivation. It is conceivable that an attacker that is willing to cause damage, is also interested in earning some money on the side. The framework should thus provide guidelines on how to deal with the calculations in the attack tree analysis for variously motivated attackers.

### C3 The framework should contain a gains parameter that is attacker independent

The previous chapter discussed that the gains parameter is currently inherently dependent on the motivation of the attacker. A way has to be found to make the gains parameter independent of the attacker's motivation to be able to analyse the attack tree for various attackers, motivated by different types of motivation. In the eventual framework a gains parameter should be used that is attacker independent.

### C4 The framework should provide guidelines to estimate the gains parameter

In the current attack tree methodology the gains parameter is very vaguely defined and no clear guidelines are given for assigning a value to the gains parameter (Buldas et al., 2006; Jürgenson &

Willemson, 2008, 2010a; Lenin et al., 2014). In order to make the gains parameter more realistic such guidelines are necessary.

## 5.2 Dilemmas

Four dilemmas have been found for which design choices are made in the design and development step. These four dilemmas are discussed in this section. The dilemmas have been distilled from the study of the state of the art in chapter 3 and from the discussion on the gains parameter in chapter 4.

### D1 Serial model or Parallel model?

During the literature study an extension was found for the attack tree methodology in which a certain order in which the elementary attacks are attempted is considered. This so called serial model is explained in detail in section 3.1.3. The serial model is an alternative to the parallel model, in which the order of the elementary attacks is not taken into account. In the parallel model the attacker is considered to attempt all of the elementary attacks at the same time (Jürgenson & Willemson, 2010b). For the current research a decision has to be made between the serial model and the parallel model. The framework could be using either of the two models.

### D2 Attack trees or Attack-Defence trees?

Another extension of the attack tree methodology found in literature is the inclusion of defence nodes in the attack tree. This results in an attack-defence tree where countermeasures that are in place can be represented in the tree (Kordy et al., 2011). The choice could be made for designing the framework in such a way that defence nodes can also be included.

### D3 Include intermediate pay-offs?

In the current attack tree methodology the gains parameter is a global one. The attacker is considered to receive the gains if s/he reaches the root node in the attack tree. In this research the gains parameter is adjusted, to make it independent of the motivation of the attacker. Also guidelines are formed for estimating the value for the gains. While changing the gains parameter it is also interesting to look at how this parameter is allocated. It may be necessary to not only be able to allocate gains to the root node, but also to intermediate nodes in the attack tree. In this case successfully performing an attack in an intermediate node will result in gains, which means that the overall gain becomes dependent on the path that an attacker takes in the attack tree. Whether or not to include such intermediate pay-offs is decided on in the design and development step.

### D4 Allow for an opt-out possibility?

Another dilemma, closely related to the dilemma on intermediate pay-offs, is whether to include an opt-out possibility for attackers in the framework. In the current attack tree methodology only attack suites are analysed that reach the root node (Buldas et al., 2006; Jürgenson & Willemson, 2008, 2010a; Lenin et al., 2014). If intermediate pay-offs exist, it might become interesting for an attacker to just try to reach some intermediate node instead of reaching the root node. During the design and development step a choice is made on whether or not to give the attacker an opt-out possibility.

# 6 Design

This chapter describes the framework that has been designed. The framework has been formed with respect to the constraints that were set up in the previous chapter. Also the dilemmas that were posed have been dealt with. How these dilemmas were dealt with is described in section 6.1. Section 6.2 describes the framework, where notions are made to show that the framework complies with the boundaries set by the constraints. In 6.3 an example is worked out to demonstrate how the new extended methodology works. The description of this design forms the answer of the fourth sub question: *How to include the motivation of attackers in the use of attack trees with regard to the requirements?*

## 6.1 Dealing with the dilemmas

In the previous chapter, four dilemmas were posed for which a design choice needs to be made. The following dilemmas were identified:

1. Serial model or Parallel model?
2. Attack trees or Attack-Defence trees?
3. Include intermediate pay-offs?
4. Allow for an opt-out possibility?

The choices made on these dilemmas are first of all based on their relation to the gains parameter. As was concluded in section 4, the gains parameter is the most important parameter to change in the design of the framework. Argumentation for the choices made on the dilemmas will therefore be made with respect to the influence they have on the gains parameter.

The same argumentation goes for both the first and the second dilemma (D1, D2). Whatever choice is made for these two dilemmas, is not considered to have a big influence on the gains parameter. When looking at the serial model as presented by Jürgenson & Willemson (2010b) the order of the attacks does not have an influence on the gain and this does not change in the framework designed. Also defence nodes do not influence the gains in the method developed by Kordy et al. (2011). The framework designed builds on the method presented by Lenin et al. (2014), and in their method the parallel model is used and no defence nodes are included. For these reasons the framework that is designed uses the parallel model and does not include the use of defence nodes. Also, in later research it will still be possible to adapt the framework, in the same way that the old methodology was altered for the serial model and for the inclusion of defence nodes.

The third dilemma is whether or not to include intermediate pay-offs (D3). Intermediate pay-offs may be necessary to be able to differentiate between the gains for certain paths taken in the attack tree. In order to illustrate why including these intermediate pay-offs may be useful, the previously used example is used. The attack tree is shown in Figure 14. In the example there are two different intermediate nodes via which an attack path can be chosen. If an attacker chooses a path via the ‘steal laptop’ intermediate node, s/he would in the end not only have obtained the secret data, but he would also have a laptop that is worth something. If an attacker chooses a path via the ‘remote access’ intermediate node, s/het would not have this. This means that a path via the ‘steal laptop’ intermediate node should result in higher gains than paths via the ‘remote access’ intermediate node. This example thus describes a case in which intermediate pay-offs would be necessary to differentiate between the gains for various attack paths. The framework designed therefore includes intermediate pay-offs.

The design choice made for the last dilemma (D4) is actually based on the decision to include intermediate pay-offs. If intermediate nodes result in gains, it might be possible that an intermediate node already results in such a high gain that an attacker will decide to stop there. The attacker would in that case not attempt to reach the root node, because that might lower the overall outcome of the

attack because of possible high costs or a low probability of success. In order to be able to deal with this in the attack tree methodology, the framework designed allows opt-out possibilities in the analysis. The attacker will thus not necessarily have to reach the reach the root node in the new framework. This concludes the design decisions made for the dilemmas posed. In the next section the designed framework is described in detail.

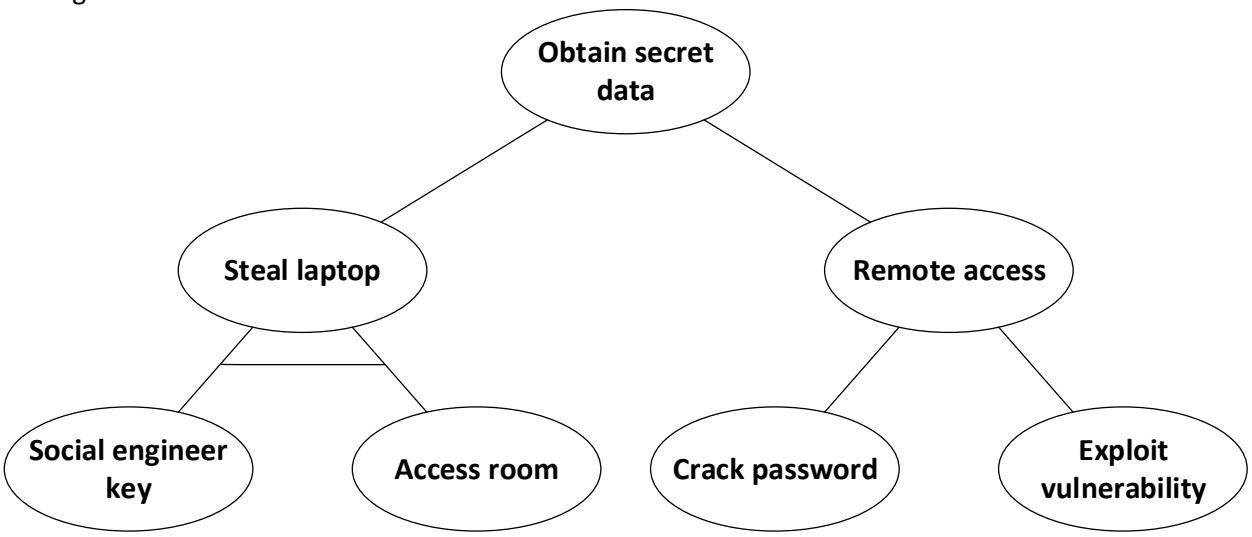


*Figure 14: Example attack tree (Based on Pieters et al., 2014)*

## 6.2 Description of the design

Based on the design choices made above and with respect to the constraints set in the previous chapter, a framework has been designed for the attack tree methodology with the inclusion of the motivation of attackers. Figure 15 (p40) is a visualization of this framework. This section will explain the steps in the framework and provide the mathematical structure.

The framework is divided into three parts. A part focusses on setting up the attack tree, another on setting up the attacker profile and the last part describes how these two parts are combined. Changes have been made in the method developed by Jürgenson & Willemson (2008, 2010a) and Lenin et al. (2014), but some aspects have stayed the same. Changes that were made are discussed more elaborately.

### 6.2.1 Setting up the attack tree

Setting up the attack tree is divided into seven separate steps. The first three steps are no different from the method developed by Lenin et al. (2014). First you have to determine what the main goal is for the attacker. This main goal will become the root node of the attack tree to create. The next step is to construct the attack tree by splitting up the root node and continue splitting up attacks until elementary level is reached for each of the nodes. These elementary attacks are then assigned values for the following parameters:

- Expenses
- Difficulty
- Required attack time
- Probability of success

The value for the expenses is a monetary one. In accordance with Lenin et al. (2014) the value for the difficulty can be *low, medium, high,* or *very high.* The possible values for required attack time are *seconds, minutes, hours* and *days.*

The fourth step of setting up the attack tree is related to the design decision to include intermediate pay-offs. In order to do so, you have to determine which intermediate nodes are considered to result

in gains. These nodes are called pay-off nodes. After determining which nodes are pay-off nodes, a value for the gains can be assigned to each of these pay-off nodes as well as the root node. In the new method, this gains parameter is split into various types of gains. The reason for doing so is related to the constraint that the framework has to deal with the various types of motivation that an attacker may have (C2). A type of gain is defined for each type of motivation. Table 9 shows what type of gain is associated with what type of motivation.

*Table 9: Motivation and their associated gain*

| Type of motivation | Type of gain |
|---|---|
| Financial benefits | Money |
| Causing damage | Damage |
| Knowledge gaining | Knowledge |
| Pleasure seeking | Pleasure |
| Notoriety within a community | Notoriety |

For each of the pay-off nodes and the root node, a value for each of the types of gains will have to be assigned. These gain values have to be attacker independent in order to meet with the constraints that were set (C3). The values for the gains will thus have to be assigned in an attacker independent way. To give a clear meaning to the attacker independent gain values, the average pay-off for an attacker is suggested as the correct value to take. With the help of weight values the pay-off can be differentiated for variously motivated attackers, where the weight value indicates how highly the attacker values the type of gain as compared to the average attacker. How to determine these weight values is discussed in section 6.2.2.

In order to comply with the constraint to include guidelines to estimate gains values within the framework, some possibilities are listed (C4). Estimating the gains for the attack tree is a very complex task, because it is hard to put an exact number on it. The fact that the designed framework uses five forms of gains makes it even more complex. Because we have to be able to compare these different types of gains with each other, it is necessary to measure each of them on the same scale. The proposed method to do so is by monetizing each type of gain. In this way, every gain is expressed in terms of money and they will therefore be comparable. Translating the money gain is in this case easy, because it is already expressed in terms of money. The damage gain could take the value of the costs the damage brings along for the defender. For the knowledge gain, you could ask yourself what the knowledge is worth to an attacker in terms of money. The same could be done for pleasure and notoriety.

The problem then still is to determine how monetary values for each of these types of gains can be estimated. Various ways can be thought of to do so. Table 10 provides a list of the most important differences between three suggested methods to determine the average pay-offs for an attacker.

*Table 10: Methods for estimating gains and their characteristics*

| | **Guesstimates** | **Human capital** | **Willingness to pay** |
|---|---|---|---|
| *Accuracy of outcome* | Low | Medium | Very |
| *Difficulty* | Medium | Very | Very |
| *Useful for what application?* | Rank outcomes | Rank outcomes | Exact outcomes |
| *Complications* | Hard to argue values | Hard for certain types of gains | Time consuming |

The first way is to use guesstimates, where the IT security expert him/herself attempts to set a value for each of the types of gains. The accuracy of the values that are set with the use of this method will probably not be very high. The IT security expert can use his/her experience or can use information

from past attacks to improve the accuracy, but still it will be hard to estimate accurate values. The main difficulty of this method lies in giving argumentation for the chosen values, which is also where complications arise. When unable to argue the values, decision makers may not trust the results that the IT security experts gets. This method of estimating the gains is most useful for ranking the outcomes of various attacks. It is probably not possible to calculate exact outcomes with this method, which means the results should not be interpreted as such. It is however possible to compare the various outcomes and in this way determine which are the more profitable attacks.

The other two ways to estimate the gain values are inspired by the way in which insurance companies put a monetary value on for example injuries or deaths. The two ways that are used to do so are based on human capital and on willingness to pay (Etter, 1987). The way to use human capital is to see how much wage is lost by the victim. The way to use willingness to pay is to look at how much a person is willing to pay to reduce the probability of injury or death. The human capital can for example be used for putting a value on notoriety, because this notoriety could lead to better job opportunities, which translates to extra wage. The willingness to pay principle can be used to determine the worth of for example the pleasure one gets from performing an attack for the average attacker.

The accuracy of the human capital method is very dependent on how fine grained the human capital is worked out. Also this human capital can we worked out better for certain types of gains as compared to others. It is for example hard to say how much extra wage an attacker gets for the pleasure gained. This is why the method is indicated as very difficult and why it is considered only to be useful for certain types of gains.

Using the willingness to pay principle, will probably result in the most accurate values for the gains. An extensive quantitative study can be performed, where attackers are asked to put a value on various results of an attack. Attackers can for example be asked to put of monetary value on the pleasure s/he receives from performing a certain attack. The average of the values these attackers give can then be assigned as the gain value. This does however require an extensive study which adds to the difficulty of, and time required for, using this method. In this way it would however be possible to get the most accurate values, in which case the outcomes can thus be interpreted as the real world values. It is however important to keep in mind that there will still be uncertainty in the outcomes, which means the accuracy of the values should not be overestimated.

From the description above it becomes clear that estimating the values for the gains is not straightforward. The three methods that are discussed each have their difficulties and shortcomings. It is however not necessary to choose one of the methods. A combination of the methods can also be used, where for each of the types of gains the most suitable method can be chosen. In the discussion chapter the issue of estimating the gains values is further discussed and recommendations for future research to overcome this issue are given.

After assigning values for the gains to each of the pay-off nodes, the Boolean formula has to be set up. In the method of Lenin et al. (2014) this formula was one function that resulted into True if the root node was reached. This Boolean formula made sure that only attack suites are analysed that satisfy the root node. A design decision was made to also allow for an opt-out possibility in the new framework, which means that the Boolean formula needs to change. In the new case, the Boolean formula consists of a set of sub-formulas. Each of these sub-formulas describes a subtree that is rooted in a pay-off node. Also there is still a Boolean formula for the root node. Within the new framework, an attack suite that satisfies just one of these sub-formulas is already seen as a satisfying attack suite. This way, the attacker does not necessarily need to satisfy the root node and thus an opt-out possibility is included. Determining these satisfying attack suites is the last step in this process.

These steps conclude the setting up of the attack tree. The result is an attack tree with parameter values assigned to the elementary nodes, described by a set of Boolean formulas of which each corresponds to a (sub) tree that has values for five types of gains associated with it. In the next section the setting up of the attacker profile is discussed.

### 6.2.2 Setting up the attacker profile

The designed framework includes four steps for setting up the attacker profile. The first step is to determine what type of attacker you want to assume. During the analysis of the state of the art described in chapter 3 it was seen that multiple types of attackers exist. There does however not seem to be consensus on the exact description of each of the types of attackers. The aim of this research has also not been to form clear attacker profiles, which means that no finite list can be presented with the possible types of attackers to include in the method described by the framework. The framework does however show which attacker characteristics have to be included in an attacker profile in order to make it suitable for the method. The allocation of values for the attacker characteristics is the second step of setting up the attacker profile. The attacker characteristics are:

- Budget
- Skill
- Time available

The budget is a monetary value. In accordance with Lenin et al. (2014) the value for the skill can be *low, medium* or *high.* The possible values for time available are *seconds, minutes, hours* and *days.*

The third step in setting up the attacker profile is assigning a weight for each type of gain. This is the step where the motivation of the attacker is taken into account in the attacker profile. By assigning weight values, the importance of each of the types of gains for the attacker can be expressed. A weight has to be assigned for the following types of gains:

- Money
- Damage
- Knowledge
- Pleasure
- Notoriety

As explained in section 6.2.1 the attacker independent gain value in the attack tree is considered the average pay-off value for an attacker. With the help of the weight value, the pay-off value for a certain attacker can be determined. The weight value for a certain type of gain thus indicates how important that type of gain is for the attacker as compared to the average attacker.

Two possible ways are discussed for choosing the weight values. The first is to freely choose values for the weights. In this way the IT security expert has the most freedom, but very high pay-off values are possible in this situation, which may not be very realistic. Also this method provides almost no guidelines to the IT security expert for allocating the values, which may make it more difficult. If the IT security expert is interested in exact values this freedom is necessary, because every restriction on what weight values to choose, may restrict him/her in getting the actual pay-off values.

It is however very hard to get the actual pay-off values and the IT security expert may thus be more interested in forming a ranking of the outcomes of the various attacks. When this computational results are preferred a more restricted method can be used for allocating the weight values. In this case the IT security expert could choose the weight values from a predefined set of values. For example only values between zero and two could be chosen. It would in this case be possible to indicate that the attacker is not interested in the type of gain at all or that he is interested in it twice as much as the average attacker and all the values in between.

The aim of this research is not to provide a definitive method of assigning the weights. It should however be understood that there are various ways of assigning values and that it possibly has an influence on what you can do with the results from the attack tree analysis. If you want the actual pay-off value for an attacker, the weight value needs to be assigned as accurate as possible, which is a complex task. Future research might aim at finding a way to include the spread of the pay-off values for the attackers in addition to the average value. Some ideas about this are described in the discussion chapter.

The last step in setting up the attacker profile is to assign a weight value for the expenses the attacker has to make. In earlier iterations of the design it was noticed that there is also a need to indicate the importance of the expenses for the attacker. When the values for the weights for the various types of gains are freely chosen, it might be possible that the influence of the expenses is completely lost if it is not possible to assign a weight to these expenses. A weight is therefore also assigned to the expenses. Assigning such a value can be explained from reality, because one attacker can find the expenses more or less important than another. An attacker that is for example very interested in damaging a certain system from an ideological perspective and has a large budget, may be less interested in the costs than an attacker with a small budget that is trying to make some money from an attack.

By performing these four steps described above, an attacker profile is set up. This attacker profile contains values for each of the attacker characteristics, a weight for each of the types of gains and a weight for the expenses. In combination with an attack tree the actual analysis can be performed in the way described in the following section.

### 6.2.3 Combining the attack tree and the attacker profile

The first two parts of the framework to set up the attack tree and to set up the attacker profile as described, can be performed completely separated from each other. In the third part the attack tree and the attacker profile are combined. The analysis of the combination of an attack tree and an attacker profile is made up of three steps. In the first step the profile satisfying attack suites are determined on the constraint basis described by Lenin et al. (2014).

After determining the attacker profile satisfying attack suites, the outcome is calculated for each of these. Here some changes are made to the mathematical structure developed by Lenin et al. (2014). The new formula for the outcome is the following:

$$Outcome_{\sigma}^{j} = Pay\_off_{\sigma}^{j} - e^{j} \sum_{X_i \in \sigma} Expenses_i$$

In this formula *j* represents the attacker profile and the pay-off is the old gains parameter, but this is now the pay-off dependent on the attacker profile. The expenses are in the new case multiplied by the weight factor for the expenses $e^j$, which is again dependent on the attacker profile. In the old case the gains parameter was an assigned value, but in the new framework it has become a formula that sums up the pay-offs of each of the pay-off nodes that are satisfied by the attack suite that is considered. The following formula is used for this:

$$Pay\_off_{\sigma}^{j} = \sum_{F(T(Y_i))(\sigma:=true)=true} P_i^j \times p_{\sigma,T(Y_i)}$$

In this formula $T(Y_i)$ represents an sub tree rooted in pay-off node $Y_i$ and $F(T(Y_i))$ represents the Boolean formula associated with this sub tree. The root node is denoted by $Y_0$. The pay-off of a certain pay-off node for a certain attacker profile is represented by $P_i^j$. The value of this is calculated by means of a utility function, where every type of gain is multiplied by the corresponding weight from the attacker profile and then summed up. The following formula is used for this summation:

$$P_i^j = \sum_{k=1}^{n} w_k^j \times g_k^i$$

In this function $w_k^j$ represents the weight for a certain type of attacker *j* for the type of gain *k* and $g_k^i$ represents the gain of type *k* for the pay-off node *i*. For now there are five types of gains which is why

the summation assumes values for *k* from 1 to 5 in the current framework. If future research points out other types of motivation, this can easily be modified.

The last step of the framework is to analyse the results and draw conclusions about the security. The security expert can at this point see how profitable certain attacks are for a certain type of attacker. Based on this s/he can decide to put in place countermeasures or not. After this first analysis, the process can be performed again, where the steps of the attack tree and the attacker profile can be performed separately from each other. This means that for a certain attacker profile, several attack trees can be analysed, but it also means that you can analyse a certain attack tree for various attacker profiles. On the next page in Figure 15 the framework is visualized.

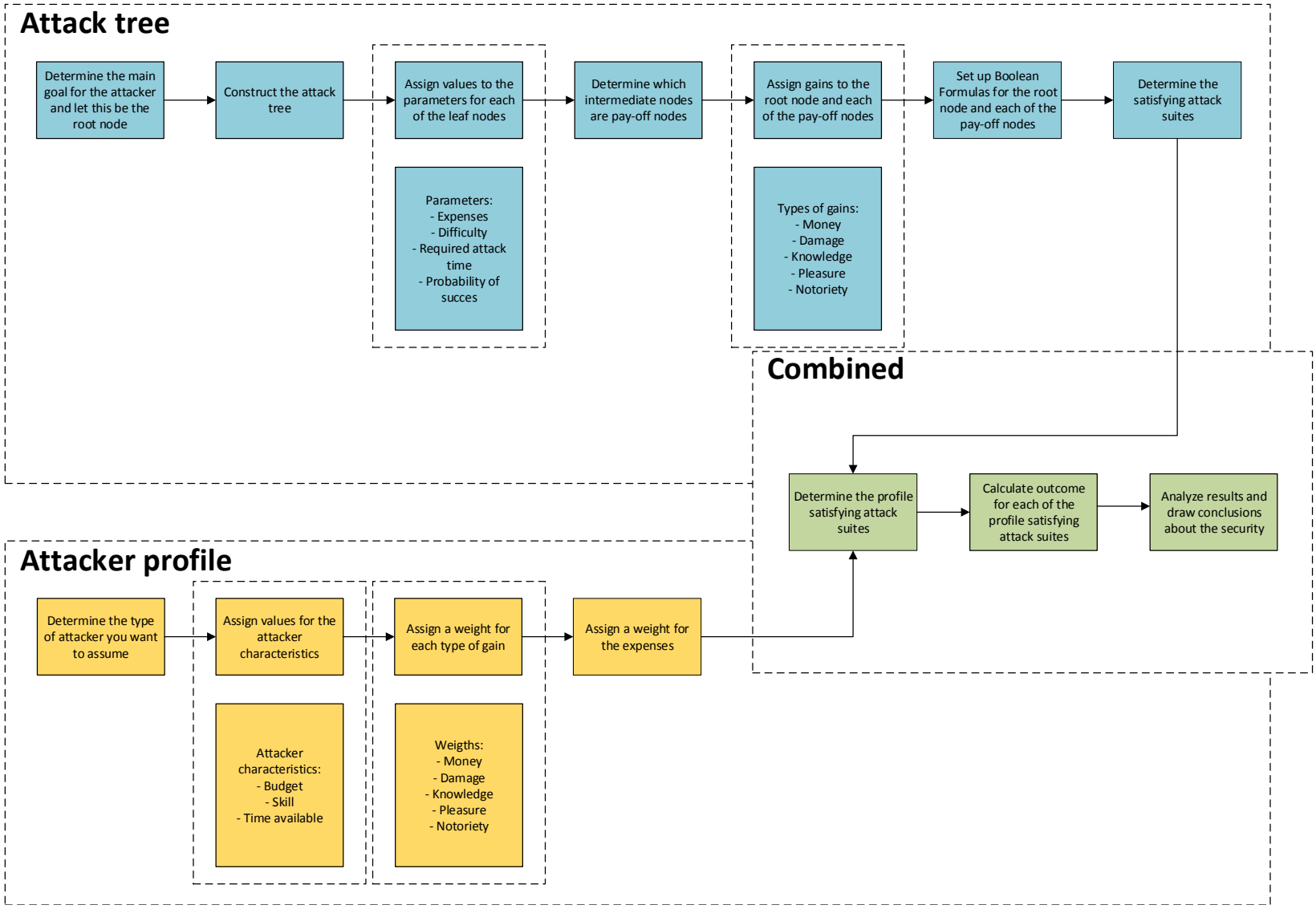


*Figure 15: Visualization of the framework*

## 6.3 Example to illustrate design

To further clarify the designed framework an example is worked out. The three parts of the framework are described in different sub sections.

### 6.3.1 Setting up the attack tree

The main goal for the attacker in this example is the same as for the example used earlier in the report, which is to obtain secret data. The attack tree is however constructed a little different than in the old example. The reason for this is that reaching one of the intermediate nodes in the earlier used example will automatically also let the attacker reach the root node. This way there would be no use for an opt-out possibility. In the new example the main goal in the root node is still obtaining secret data. The way to do so is by stealing a laptop AND decrypting a laptop. For stealing a laptop the attacker still has to social engineer a key AND access a room. Decrypting the laptop can either be done by obtaining the encryption key OR using brute force. The attack tree that corresponds to this is shown in Figure 16. Indicators for each of the nodes have been added, which are used from now on.

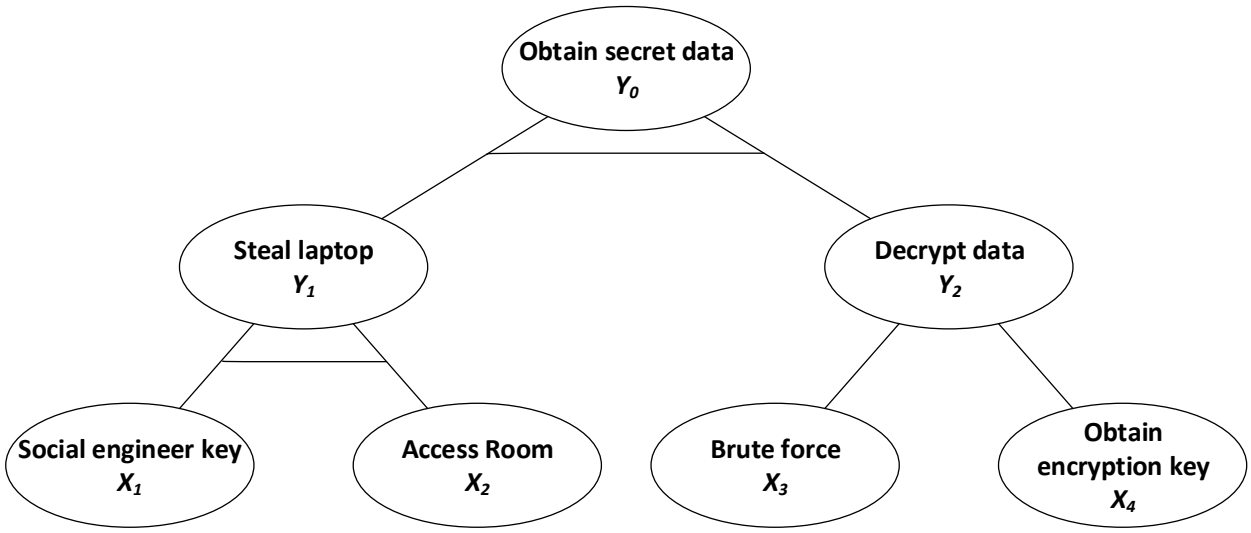


*Figure 16: Attack tree for example to explain framework*

The next step in the framework is to assign parameter values to each of the elementary attacks. These values are presented in Table 11.

*Table 11: Parameter values of elementary attacks*

| Elementary attack | Expenses | Difficulty | Time needed | Probability of success |
|---|---|---|---|---|
| $X_1$ | 100 | Low | Seconds | 0.5 |
| $X_2$ | 200 | Medium | Minutes | 0.8 |
| $X_3$ | 400 | Medium | Hours | 0.8 |
| $X_4$ | 250 | Medium | Minutes | 0.4 |

Now it has to be determined which intermediate nodes are pay-off nodes. In this example the root node is a pay-off node as well as $Y_1$. For both of these pay-off nodes a value has to be assigned for each of the types of gains. For the sake of simplicity, just two types of gains are used in this example, which are money and knowledge. The values that are assigned are presented in Table 12.

*Table 12: Gains values of the pay-off nodes*

| Pay-off node | Money gain | Knowledge gain |
|---|---|---|
| $Y_0$ | 500 | 500 |
| $Y_1$ | 1000 | 0 |

After assigning these values the set or Boolean formulas is set up for the attack tree. For this example attack tree the following Boolean formulas apply:

$$F(T(Y_0)) = (X_1 \wedge X_2) \wedge (X_3 \vee X_4)$$
$$F\big(T(Y_1)\big) = X_1 \wedge X_2$$

The last step for setting up the attack tree is to determine the satisfying attack suites. For the example attack three the satisfying attack suites are {$X_1$, $X_2$}, {$X_1$, $X_2$, $X_3$}, {$X_1$, $X_2$, $X_4$} and {$X_1$, $X_2$, $X_3$, $X_4$}. This concludes the setting up of the attack tree.

### 6.3.2 Setting up the attacker profile

In order to show the influence of the weight values, two attacker profiles are set up. For both of these attacker profiles the analysis is performed. The first step in setting up an attacker profile is to determine what type of attacker you want to assume. In this example both the attackers are assumed to be highly skilled attackers with lots of time available. The next step is to assign values for the attacker characteristics. The values are assumed the same for both attackers and are summarized in Table 13.

*Table 13: Attacker characteristics values for the attacker profile*

| Attacker characteristic | Profile value |
|---|---|
| **Budget** | 1000 |
| **Skill** | High |
| **Time available** | Days |

The last two steps of setting up the attacker profile are to assign weights for each type of gain and to assign a weight value for the expenses. Because only two types of gains are assumed for the example, only weights are assigned for these two types of gain. The values of the weights for the types of gains and the weight for the expenses are both shown in Table 14. This completes the attacker profiles. The next section describes how the attacker profiles get combined with the attack tree.

*Table 14: Weight values for the attacker profile*

| Weight for… | Weight value for attacker 1 | Weight value for attacker 2 |
|---|---|---|
| **Money** | 2 | 0.5 |
| **Knowledge** | 0.5 | 2 |
| **Expenses** | 1.5 | 1.5 |

### 6.3.3 Combining the attack tree and the attacker profile

The first step in combining the attack tree and the attacker profiles is to determine the profile satisfying attack suites. In the example the attackers that are assumed are able to perform each of the attack suites, which means that the profile satisfying attack suites are {$X_1$, $X_2$}, {$X_1$, $X_2$, $X_3$}, {$X_1$, $X_2$, $X_4$} and {$X_1$, $X_2$, $X_3$, $X_4$}.

Next the calculations are performed using the following formulas:

$$Outcome_{\sigma}^{j} = Pay\_off_{\sigma}^{j} - e^{j} \sum_{X_i \in \sigma} Expenses_i$$

$$Pay\_off_{\sigma}^{j} = \sum_{F(T(Y_i))(\sigma := true)=true} P_i^j \times p_{\sigma,T(Y_i)}$$

$$P_i^j = \sum_{k=1}^{5} w_k^j \times g_k^i$$

The results of this are shown in Table 15 for attacker 1 and in Table 16 for attacker 2. There is a clear difference between the outcomes for each of the attackers. In this case you could conclude that it is not interesting for attacker 2 to try to obtain the secret data, because none of the attack suites results in a positive outcome. For attacker 1 however, there are multiple attack suites that have a positive outcome. The outcome of the attack suite in which the opt-out possibility is used is the highest, which could thus be considered as the most likely attack to be attempted by the attacker.

*Table 15: Satisfying attack suites and the calculated outcome for attacker 1*

| Attack suite (σ) | Pay-off ($Y_0$) | Pay-off ($Y_1$) | $P_{succes}(Y_0)$ | $P_{succes}(Y_1)$ | Weighted expenses | Outcome |
|---|---|---|---|---|---|---|
| $X_1, X_2$ | 1250 | 2000 | 0 | 0.4 | 450 | 350 |
| $X_1, X_2, X_3$ | 1250 | 2000 | 0.32 | 0.4 | 1050 | 150 |
| $X_1, X_2, X_4$ | 1250 | 2000 | 0.16 | 0.4 | 825 | 175 |
| $X_1, X_2, X_3, X_4$ | 1250 | 2000 | 0.352 | 0.4 | 1425 | -185 |

*Table 16: Satisfying attack suites and the calculated outcome for attacker 2*

| Attack suite (σ) | Pay-off ($Y_0$) | Pay-off ($Y_1$) | $P_{succes}(Y_0)$ | $P_{succes}(Y_1)$ | Weighted expenses | Outcome |
|---|---|---|---|---|---|---|
| $X_1, X_2$ | 1250 | 500 | 0 | 0.4 | 450 | -250 |
| $X_1, X_2, X_3$ | 1250 | 500 | 0.32 | 0.4 | 1050 | -450 |
| $X_1, X_2, X_4$ | 1250 | 500 | 0.16 | 0.4 | 825 | -425 |
| $X_1, X_2, X_3, X_4$ | 1250 | 500 | 0.352 | 0.4 | 1425 | -785 |

## 6.4 Conclusion

In this chapter the framework that is designed has been described in detail. Also an example was worked out to further clarify the working of the framework. The next step is to validate the framework, which is described in the following chapter.

# 7 Framework validation

In the previous chapter the designed framework was discussed, the framework however needs to be validated. This chapter describes the steps that were performed to validate the framework and this gives an answer to the fifth sub question: *Does the method add the expected value?* As can be seen from the sub question, the focus in this chapter does not lie on the mathematical structure of the framework. Validation of the framework is done by arguing whether the designed framework actually provides the prospected added value. This added value was described in chapter 4 and can be summarized as follows:

- The gains parameter is made independent of the type of attacker
- Various pay-offs are possible for variously motivated attackers
- The gains parameter is made more realistic

There are various ways to validate the framework of which a list is given by Hevner (2004). The ideal way to validate the designed framework would be by means of a field study, where the predictive capability of the framework would be tested. Within the time frame of this research it was however not possible to perform such a field study. Section 7.1 describes how such a field study would have to be performed if it would have been possible.

The validation of the framework that has been performed, is done in two different ways. First an example attack tree on I-voting is used as a case study. This case study is used to show how the framework can be used on a real world case. Also it is indicated where the various points of added value can be noticed while working out the case study. The case study is described in section 7.2. Also an interview with an expert was performed to validate the model, which adds to the reliability of the framework. The way in which the framework was expert validated is described in section 7.3. Section 7.4 provides a reflection on the performed validation.

## 7.1 Field study to prove the prediction capability of the framework

The framework is intended to give the IT security experts a tool to predict the outcome for an attacker for various attacks. With including the motivation of attackers in this framework, the outcome is expected to become more realistic as the gains parameter is made more realistic. The best way to prove this is actually the case, is by using the framework to predict the outcome and comparing this predicted value to the actual real world value of the outcome. Two problems exist with performing such a field study.

The first problem is that performing such a field study takes a lot of time and effort. First a real world system would need to be sought on which the field study can be performed. For this system a complete attack tree needs to be set up and for each of the nodes the parameters need to be allocated. This process takes more time than available within the timeframe of this research.

Next to forming the attack tree, multiple attackers need to be sought that are interested in attacking the system. Each of these attackers needs to be willing to fully explain the value s/he gives to the outcome of the attack. Also s/he needs to be able to do so, which may also be a challenge. This process is also too consuming within the time frame of this research, which is why alternative forms of validation were used. The following sections describe the validation that was performed.

## 7.2 I-voting case study

Cybernetica was found willing to provide an attack tree to use as a case study for validating the framework. The attack tree is used for analysing attacks in an Internet voting (I-voting) environment. The attack tree is formed based on the master thesis of Torn (2014), who based his structure on the description of modelling threats of a voting method written by Heiberg & Willemson (2014).

I-voting is used in Estonia as a complement to traditional voting methods. Voters are in this way able to cast their vote via the internet. The attack tree describes attacks that an attacker can perform in order to alter the outcome of the elections through this I-voting environment. Two types of attacks are considered in the attack tree, which are manipulation attacks and revocation attacks.

Manipulation attacks are those attacks that aim to alter the result of votes that have already been cast by voters. The attacker attempts to remove already cast votes from the system in case of a revocation attack. In order to deal with the scale of the attack tree a choice has been made to focus on the manipulation attacks. A table describing the manipulation attacks part of the attack tree is found in appendix B. No visualization is provided, because the attack tree is too big.

In different sub sections, a description is given of the application of the framework, where notions are made if the prospected added value is reached. The added value of this case study as compared to the example used in the previous chapter is that the attack tree describes a real world case. Values for the parameters are used from the attack tree that was provided. The case study shows how the motivation of the attacker can change the outcome for the attacker. Also these outcomes are compared to the outcomes that resulted from using the original method presented by Lenin et al. (2014).

### 7.2.1 Setting up the attack tree for the I-voting case

The first three steps of the framework, in which the attack tree is constructed and parameter values are assigned for each of the elementary attacks, are already performed. The next step is to determine which nodes are intermediate pay-off nodes. By going through the attack tree a part of the attack tree was chosen as a focus point, because intermediate pay-offs were considered to exist in this part. The focus point are the attacks that lead to the 'Fake voting applications' attack. This part of the tree is visualized in Figure 17. The 'Fake voting applications' node is connected to the root node via an OR node. The root node is also an OR node, so if the 'Fake voting application' node is satisfied, the root node is also satisfied. In the visualization, elementary attacks have been coloured green and pay-off nodes have been coloured blue. The nodes that are considered pay-off nodes are the 'Fake voting applications' node, the 'Replace app on NEC web server' node and the 'From official appstore' node. Because reaching the 'Fake voting application' node also means reaching the root node, this is considered a pay-off node. The other pay-off nodes are determined by thinking about what part of the attack already brings some results. If an attacker manages to replace an app on the National Election Committee (NEC) web server it is likely that there would for example already be some notoriety gain associated with it. This can already be noticed by an attacker community, which would be proof of the attacker performing the attack. The same reasoning goes for the distribution of a fake verification app through the official appstore. The three nodes that will thus be assigned pay-offs are 'Fake voting application' ($Y_0$), 'Replace app on NEC web server' ($Y_6$) and 'From official appstore' ($Y_7$).

The next step is to assign gains to each of the pay-off nodes. For this case, two types of gains are used for the sake of simplicity. These are different from the example used in the previous chapter and are the two types of gains associated with the motivations that are considered most applicable to the I-voting case. The first is damage, because an attacker might be interested in manipulating the voting process to cause damage to the participating parties. The second type of gain is notoriety, because manipulating such a public event as a voting process may be noticed by an attacker community. In the current attack tree the gains for reaching the root node is set to various values (Torn, 2014). In this case the value of 10M is assumed for the overall attack tree. This same number is kept for the total of all the gains in the new model. The values assigned for the gains are shown in Table 17. These gains have now been assigned independent of any type of attacker. In a later stage an attacker profile is assumed that can value each of these types of gains differently. The first point of the prospected added value is thus realized by the designed framework.

*Table 17: Gains values of the pay-off nodes*

| Pay-off node | Damage gain | Notoriety gain |
|---|---|---|
| $Y_0$ | 7M | 1M |
| $Y_6$ | 0.5M | 0.5M |
| $Y_7$ | 0.5M | 0.5M |

The parameter values of each of the elementary attacks in the part of the attack tree used for the validation are shown in Table 18.

*Table 18: Parameter values for the elementary attacks*

| Elementary attack | Expenses | Difficulty | Time needed | Probability of success |
|---|---|---|---|---|
| $X_1$ | 200940 | Medium | Hours | 0.95 |
| $X_2$ | 200940 | Medium | Hours | 0.95 |
| $X_3$ | 1400 | Medium | Hours | 0.95 |
| $X_4$ | 24470 | Medium | Hours | 0.001 |
| $X_5$ | 1389700 | Medium | Hours | 0.005 |
| $X_6$ | 16825 | Medium | Hours | 0.002 |
| $X_7$ | 2552880 | Medium | Hours | 0.33 |
| $X_8$ | 2552880 | Medium | Hours | 0.33 |
| $X_9$ | 2498460 | Medium | Hours | 0.005 |
| $X_{10}$ | 26430 | Medium | Hours | 0.05 |
| $X_{11}$ | 2552880 | Medium | Hours | 0.33 |
| $X_{12}$ | 2552880 | Medium | Hours | 0.33 |
| $X_{13}$ | 2498460 | Medium | Hours | 0.005 |
| $X_{14}$ | 26430 | Medium | Hours | 0.00001 |
| $X_{15}$ | 0 | Medium | Hours | 0.005 |

After constructing the attack tree and determining the pay-off nodes, the set of Boolean formulas can be formed. This attack tree is described by the following set:

$$F(T(Y_0)) = (X_1 \wedge X_2) \wedge \big((X_3 \wedge (X_4 \vee X_5 \vee X_6)\big) \vee (X_7 \vee X_8 \vee X_9)) \wedge \big((X_{10} \wedge (X_{11} \vee X_{12} \vee X_{13})\big) \vee (X_{11} \vee X_{12} \vee X_{13})) \wedge X_{14}$$

$$F\big(T(Y_6)\big) = X_7 \vee X_8 \vee X_9$$

$$F(T(Y_7)) = X_{10} \wedge (X_{11} \vee X_{12} \vee X_{13})$$

The last step in constructing the attack tree is setting up the satisfying attack suites. Within the part of the attack tree used for the validation there are a lot of satisfying attack suites, which makes it hard to consider each of them when performing the analysis by hand. For this reason, three attack suites were chosen to use for the analysis. These attack suites are {$X_1$, $X_2$, $X_3$, $X_4$, $X_{14}$, $X_{15}$}, { $X_1$, $X_2$, $X_7$, $X_{14}$, $X_{15}$} and {$X_1$, $X_2$, $X_7$, $X_{10}$, $X_{11}$, $X_{15}$}. The reason for choosing these is that they each satisfy a different number of the Boolean formulas, which means that the gains of the attack suites will differ.


*Figure 17: Visualization of attack tree for validation*

### 7.2.2 Setting up the attacker profile for the I-voting case

For the I-voting case, two types of attackers are assumed to show that the prospected added value of being able to have different gains for variously motivated attackers, is reached. The attackers are considered to have a large enough budget to perform the three attack suites that were set up. Also their skill level is considered high enough and they are considered to have enough time available to perform each of the attack suites. One attacker is assumed that is mainly motivated by causing damage and another that is mainly motivated by gaining notoriety. Both of the attackers are considered not to care much about expenses. The weights chosen for the attackers are shown in Table 19.

*Table 19: Weight for the attackers in the I-voting case*

| Weight for… | Weight value for attacker 1 | Weight value for attacker 2 |
|---|---|---|
| **Damage** | 0.75 | 1.5 |
| **Notoriety** | 1.5 | 0.75 |
| **Expenses** | 0.5 | 0.5 |

### 7.2.3 Combining the attack tree and the attacker profiles for the I-voting case

Now that the attack tree and the attacker profiles are set up, the outcome for each of the attackers for each of the attack suites can be calculated. The results of this are shown in Table 20 for attacker 1 and in Table 21 for attacker 2. In order to compare the results of the new framework with the results from the method presented by Lenin et al. (2014), Table 22 has been added that shows the outcomes for the chosen attack suites based on the old model. Large changes are noticeable in the outcomes between the new framework and the old model. This is mainly because the expenses are considered less important by the attackers. These expenses have a large influence on the outcome because of the low probability of success values that lower the pay-offs.

When comparing the outcomes for the two different attackers it can be seen that only for one of the three attack suites the outcome differs. This is also a result of the low probabilities of success, which diminishes the effect of the pay-off values. The third attack suite does however have a higher probability of success, and here it can be seen that the outcome is higher for an attacker mainly motivated by causing damage than for an attacker mainly motivated by gaining notoriety.

What can clearly be seen by this worked out case, is that the pay-off for the variously motivated attackers is different for each of the attack suites. This satisfies the second point of the prospected added value of the designed framework. The last point of the prospected added value of the framework, which is about making the gains parameter more realistic is a lot harder to proof. The values for the gains are still estimates or at best guesstimates. The inclusion of intermediate pay-off nodes and the splitting up of the gains parameter in various types of gains, do however add to the flexibility of the gains parameter. This flexibility gives the IT security experts the opportunity to define the gains parameter in more detail and thus more realistically.

*Table 20: Attack suites and the calculated outcome for attacker 1*

| Attack suite ($\sigma$) | Pay-off ($Y_0$) | Pay-off ($Y_6$) | Pay-off ($Y_8$) | $P_{succes}$ ($Y_0$) | $P_{succes}$ ($Y_6$) | $P_{succes}$ ($Y_7$) | Weighted expenses | Outcome |
|---|---|---|---|---|---|---|---|---|
| $X_1, X_2, X_3, X_4, X_{14}, X_{15}$ | 6.75M | 1.125M | 1.125M | $4.3E^{-11}$ | 0 | 0 | 227090 | -227090 |
| $X_1, X_2, X_7, X_{14}, X_{15}$ | 6.75M | 1.125M | 1.125M | $1.5E^{-8}$ | 0.33 | 0 | 1490595 | -1119345 |
| $X_1, X_2, X_7, X_{10}, X_{11}, X_{15}$ | 6.75M | 1.125M | 1.125M | $2.5E^{-5}$ | 0.33 | 0.0165 | 2767035 | -2377054 |

*Table 21: Attack suites and the calculated outcome for attacker 1*

| Attack suite (σ) | Pay-off ($Y_0$) | Pay-off ($Y_6$) | Pay-off ($Y_8$) | $P_{succes}$ ($Y_0$) | $P_{succes}$ ($Y_6$) | $P_{succes}$ ($Y_7$) | Weighted expenses | Outcome |
|---|---|---|---|---|---|---|---|---|
| $X_1, X_2, X_3, X_4, X_{14}, X_{15}$ | 11.25M | 1.125M | 1.125M | $4.3E^{-11}$ | 0 | 0 | 227090 | -227090 |
| $X_1, X_2, X_7, X_{14}, X_{15}$ | 11.25M | 1.125M | 1.125M | $1.5E^{-8}$ | 0.33 | 0 | 1490595 | -1119345 |
| $X_1, X_2, X_7, X_{10}, X_{11}, X_{15}$ | 11.25M | 1.125M | 1.125M | $2.5E^{-5}$ | 0.33 | 0.0165 | 2767035 | -2376941 |

*Table 22: Attack suites and the calculated outcome based on old method*

| Attack suite (σ) | $P_{success}$ | Expenses | Outcome |
|---|---|---|---|
| $X_1, X_2, X_3, X_4, X_{14}, X_{15}$ | $4.3E^{-11}$ | 454180 | -454180 |
| $X_1, X_2, X_7, X_{14}, X_{15}$ | $1.5E^{-8}$ | 2981190 | -2951190 |
| $X_1, X_2, X_7, X_{10}, X_{11}, X_{15}$ | $2.5E^{-5}$ | 5534070 | -5533820 |

## 7.3 Expert validation

The second part of the validation is based on the opinion of experts about the designed framework. The idea was to interview various experts to get a general overview of the validity of the framework. Due to time and resources constraints it was however only possible to interview one expert on the validity of the framework. Barbara Kordy was interviewed to discuss the current framework. The main findings were that the added value of the designed framework is that it is possible to represent the gains in a more fine grained way. With the intermediate pay-offs it is possible to better indicate where the gains come from. A point of critique however was that the use of an opt-out possibility is undermining the main purpose of using attack trees. The root node in attack trees usually indicates the goal that the attacker is trying to reach and in this case it would thus be unusual to assume that an attacker would also settle for reaching an intermediate node. If this is not considered a problem however, the opt-out possibility can still be used.

Another way in which experts could have been included in the validation, is by asking them to set a value for the outcome for a certain attack and attacker combination without using the mathematical structure of the framework. In parallel the mathematical structure could be used to also predict the outcome of the attack and attacker combination. By comparing these values the validity of the framework could be checked. In the best case, multiple experts are asked to do so, because their estimates will also vary. By taking an average of these estimated values, a fairly correct value could be determined. This was also not possible due to time and resources constraints. This could thus in the future be carried out to further prove the validity of the framework.

## 7.4 Reflection on the validation

The validation of a quantitative framework as designed in this research is a complex and time consuming process. This section presented a few possibilities that can be used to do so, like performing a field study or consulting experts. Time and resource constraints let however to a diminished validation of the framework. A case study is used to present the way in which the framework can be used on a real world case. Also one expert was interviewed on the validity of the model. In the future more effort could be put in further validating the framework.

# 8 Conclusions

This report describes the process that was gone through to design a framework for the attack tree methodology in which the motivation of attackers is included. With the help of this framework IT security experts can analyse the attack tree for variously motivated attackers. The main question this research sought an answer for was the following: *How can the motivation of attackers be included in the use of attack trees for cyber threat analysis?*

This chapter first summarizes the answers to the sub question. Based on the answers from the sub questions, an answer is formulated to the main research question. Section 8.1 describes the summarized answers to the sub questions and in section 8.2 the answer to the main question is given.

## 8.1 Answers to the sub questions

The aim of the chapters 3 through 7 was to give an answer to the five sub questions that were formulated at the start of the research. The answers to these questions are summarized in this section. The following questions were set up:

- $Q_1$: What is the current state of the art regarding:
  - attack trees?
  - attacker profiling, with a special focus on motivation?
  - the combination of attacker profiles and attack trees?
- $Q_2$: What value could the inclusion of motivation in the use of attack trees add to the information gained from the attack tree analysis?
- $Q_3$: What are the requirements for a framework that includes the motivation of attackers in the use of attack trees?
- $Q_4$: How to include the motivation of attackers in the use of attack trees with regard to the requirements?
- $Q_5$: Does the method add the expected value?

### The current state of the art

The attack tree methodology has been around for a while to analyse complex attacks where multiple attack paths are possible (Pieters & Davarynejad, 2014). Weiss (1991) was the first to introduce the methodology, but its name was later introduced by Schneier (1999). Over time the methodology has been improved by including different parameters and by developing several extensions for the attack trees. Also effort has been put into making the parameters in the attack tree attacker independent in order to make the attack tree reusable for analysing multiple types of attackers without having to update the parameter values.

Within the attack tree methodology, already a way to include the skill and the resources of the attacker has been developed by Lenin et al. (2014). In this method it is however not possible yet to include the motivation of the attacker. From studying the state of the art of the attacker profiles, which are descriptions of various types of attackers, it was concluded that motivation is also an important attacker characteristics that is assumed to have an influence on the way the attacker behaves. The different types of motivation that were found, are:

- Financial benefits
- Causing damage
- Knowledge gaining
- Pleasure
- Notoriety

The research gap that this research tried to resolve is the inclusion of this attacker motivation in the attack tree methodology.

### The prospected added value

In order to be able to analyse the attack tree for variously motivated attackers, the parameters of the attack tree have to be made independent of this attacker motivation. After analysing the possible influence of this attacker characteristic on the parameters, it was concluded that the main parameter to change in the current methodology is the gains parameter. In the current methodology the gains parameter is a global parameter, which is considered the same for each type of attacker and for each way in which the attack is performed.

In altering this parameter the biggest possible added value was found. The three ways in which the gains parameter can be improved by including the motivation of attackers in the attack tree methodology are:

- The gains parameter should be independent of the type of attacker
- Various gains should be possible for variously motivated attackers
- The gains parameter should become more realistic

### The list of requirements

For the design of the framework, two kinds of requirements were set up, which are constraints and dilemmas. The constraints are those requirements that the design has to comply with and the dilemmas describe two options for which a decision is made in the design and development phase.

Based on the analysis of the current methods and with the objective of the design in mind, the following four constraints were set up:

1. The framework should use multi-parameter attack trees
2. The framework should be able to deal with the various types of attacker motivation
3. The framework should contain a gains parameter that is attacker independent
4. The framework should provide guidelines to estimate the gains parameter

Next to these four constraints, there were also four dilemmas to deal with. The dilemmas are formulated as questions and are the following:

1. Serial model or Parallel model?
2. Attack trees or Attack-Defence trees?
3. Include intermediate pay-offs?
4. Allow for an opt-out possibility?

### The designed framework

Based on the list of requirements a framework was designed in which the choice was made to use the serial model, use attack trees, include intermediate pay-offs and include an opt-out possibility. The framework is build up of three parts which are the setting up of the attack tree, the setting up of the attacker profile and the analysis of the two combined. The framework describes what parameters to include in the attack tree and what attacker characteristics to include in the attacker profile. A mathematical structure is provided to calculate the outcomes for the combination of an attack tree and an attacker profile. The framework has been demonstrated by means of a worked out example.

### The validation

The designed framework has been validated by means of a case study on an I-voting attack tree and by means of an expert validation. With the help of the attack tree on I-voting it was shown that the framework does make the gains parameter independent and that variously motivated attackers can have various pay-offs without having to alter any parameters in the attack tree.

It was harder to validate whether the framework makes the gains parameter more realistically. It was however concluded that the splitting up of the gains and the inclusion of intermediate pay-offs, adds to the flexibility of the IT security expert in assigning values for the gains. Methods have been described that can be used to further validate the model.

## 8.2 Answer to the main question

The previous section described the summarized answers to the sub questions. The combined answers helped to find an answer to the main research question: *How can the motivation of attackers be included in the use of attack trees for cyber threat analysis?* This section provides this answer by focussing on the changes that had to be made to the old attack tree methodology.

In order to include the motivation of attacker, the system properties and the attacker characteristics need to be further separated from each other. Within the designed framework a clear distinction is made between setting up the attack tree and setting up the attacker profile. Within the process of setting up the attack tree, parameter values are assigned that reflect the system properties. The process of setting up the attacker profiles focusses on the attacker properties.

The main changes made in the attack tree methodology as opposed to previous methods, can be found in the gains parameter. In order to deal with the various types of motivation, the gains parameter has also been split into various types of gains. Every type of motivation an attacker may have, has an associated gain.

Another change to the gains is how it is translated to the actual pay-off for the attacker. In the old methods, the gains was supposed to be valued the same by each type of attacker, which meant that the pay-off for the attacker was equal to the gains. This pay-off is however in reality dependent on the motivation of the attacker. If an attack only gains money, an attacker motivated by notoriety values that gain lower than an attacker that is motivated by financial benefits. The pay-off for an attacker motivated by notoriety should thus be lower than the pay-off of an attacker motivated by financial benefits. The way in which this has been included in the framework is by changing the gains into a utility function that results in the pay-off for a certain type of attacker. Within the attacker profile weights are assigned for every type of gain, in which the motivation of the attacker can be reflected. By using these weights, an attacker motivated in multiple ways can also be analysed with the help of the framework.

Also in the process of making the gains parameter more realistic and to allow the IT security expert to better define gains within the attack tree, the possibility of intermediate pay-offs was included in the framework. With these intermediate pay-offs, gains do not necessarily have to be assigned to the root node of the attack tree, but can also be assigned to other nodes. This adds to the flexibility of the gains parameter within the attack tree. Also the gain becomes path dependent, which is a good thing, because performing an attack in one way, may result in higher gains than performing an attack in another. It is for example likely that stealing a laptop to obtain secret data results in more gains than obtaining the data via remote access, because the laptop itself is also worth something.

This concludes the answer to the main research question. The new framework for the attack tree methodology is considered to be an upgrade from the older methods, but it still has some drawbacks and possibilities for improvement exist. In the next chapter these drawbacks are discussed and some recommendations for further research are provided.

# 9 Discussion & Recommendations for future research

In the framework as it is presented in this report there is still room for improvement. A few of the shortcomings of the framework are discussed here. Also some recommendations for future research are linked to these shortcomings.

### Overall usefulness of the framework

The framework in its current form is most useful for IT security experts that are interested in comparing the outcomes that various attackers may have for attacking their systems. With a relatively low number of values, the IT security expert is able to form an overview of the profitableness of various attacks for various attackers. Exact values for the outcomes are however hard to determine.

The framework is based on the method as developed by Lenin et al. (2014) because of the context in which this research was performed. Possible shortcomings of this method are thus inherited by the designed framework. In section 4.1 already some dependencies between parameters were mentioned that are neglected in the method. One additional dependency is important to notice, which is not between an attacker and an attack characteristic, but between two attack characteristics. In the framework the difficulty and the probability of success of an attack are considered independently. These parameters do however seem to be very dependent on each other as a difficult attack is less likely to be successful than an easy attack. Future research may look into the interrelatedness of these two parameters.

### Consistency with Mauw & Oostdijk framework

As was stated during the state of the art, the attack tree methodology was formalized by Mauw & Oostdijk (2006). Even though the framework designed is based on the method by Lenin et al. (2014), which is consistent with the formalization of Mauw & Oostdijk, it is not sure whether the designed framework is also consistent with this formalization. Checking whether this consistency exists did not fall within the scope of this research project, but is definitely something that can be done in future research. Probably this can be combined with the next point of discussion.

### Software implementation of the framework

The designed framework uses different formulas for calculating the outcome and also uses more than one Boolean formula for which satisfying attack suites need to be found. These changes lead to the need for altering the algorithms used for analysing attack trees with the help of software. This software implementation of the framework will be necessary, because doing the analysis by hand gets too computationally heavy very fast with larger attack trees. Future research could aim at automating the calculations used in the designed framework.

### Estimating values for the gains

The current framework still relies on estimations of the values for the gains, made by the IT security expert. Getting these estimations right is a very complex task or might even be impossible. Therefore in future research it might be fruitful to look for methods in which no single value is needed for the gains. Instead you would for example be able to include interval values. Jürgenson & Willemson (2007) already present a method in which some parameters can be taken as interval values. This may serve as a starting point for doing so for the gains parameters as well.

Another possibility to deal with the uncertainty of the gains values is to include a sensitivity analysis. In this way it could be checked whether the results will change a lot when changes are made to the estimated values. Future research could focus on implementing such a sensitivity analysis in the current framework.

### Taking the spread of the pay-offs into account

In section 6.2.1 various ways to estimate the attacker independent gain values were described, where the attacker independent gain was suggested to be the average pay-off value for an attacker. When

applying the willingness to pay principle, you may use pay-off values from many attackers to calculate this average value. Later on the attacker is assigned a weight value, with which the pay-off for a certain type of attacker is calculated. In this process you lose information about the spread that can be found in the pay-off values presented by the attackers when determining the average pay-off value. In future research a way can be sought in which this spread can be included in the calculation of the actual pay-off values.

A possible way to do so is by including the standard deviation in calculating the pay-off value for a certain attacker. The weight value would in this case not be multiplied by the independent gain value, but by the standard deviation. In this way the weight value represents the number of standard deviations the pay-off for a certain attacker lies from the average pay-off value. The formula for the pay-off value for a certain pay-off node for a certain attacker would in this case be the following:

$$P_i^j = \sum_{k=1}^{n} \mu_{g_k^i} + sd_{g_k^i} \times w_k^j$$

### Make the probability of success independent of the attacker

In the designed framework the probability of success is not made independent of the attacker yet. Lenin et al. (2014) already describe a way in which the probability of success might be influenced by the skill of the attacker. In this research the motivation in terms of the drive of the attacker was also mentioned. It is likely that an attacker that has a bigger drive for committing an attack will keep trying, which might influence the probability of success. Future research could investigate the possibilities of making the probability of success parameter in the attack tree independent of the attacker.

### Other modelling methods to include attacker motivation in cyber threat analysis

The current research merely focusses on the use of attack trees for cyber threat analysis. In future research it may also be interesting to look at including the attacker motivation in other methodologies. One type of model that could also be used is an influence diagram. An influence diagram is an augmented Bayesian network that includes “decision variables, representing decision options and utility functions, representing preferences” (Kjaerulff & Madsen, 2008, p13). Influence diagrams could thus be used to model the decisions an attacker may take and the states of nature that can result from that. In these influence diagrams utility functions can be used to represent the pay-offs for an attacker.

# Literature


Barber, R. (2001). Hackers profiled—who are they and what are their motivations?. *Computer Fraud & Security*, *2001*(2), 14-17.

Blau, J. (2004). Viruses: From Russia, With Love?. *PC World.* Retrieved from: http://www.pcworld.com/article/116304/article.html

Buldas, A., Laud, P., Priisalu, J., Saarepera, M., & Willemson, J. (2006). Rational choice of security measures via multi-parameter attack trees. In*Critical Information Infrastructures Security* (pp. 235-248). Springer Berlin Heidelberg.

Casey, T. (2007). Threat Agent Library Helps Identify Information Security Risks. *Intel White Paper, September*.

Casey, T. (2015). Understaning Cyberthreat Motviations to Improve Defense. *Intel White Paper.*

Edge, K. S., Dalton, G. C., Raines, R. A., & Mills, R. F. (2006, October). Using attack and protection trees to analyze threats and defenses to homeland security. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*(pp. 1-7). IEEE.

Etter, I. B. (1987). The National Safety Council's estimates of injury costs. *Public Health Reports (1974-)*, 634-636.

Greenberg, A. (2012). Shopping For Zero-Days: A Price List For Hackers' Secret Software Exploits. *Forbes.* Retrieved from: http://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-days-an-price-list-for-hackers-secret-software-exploits/

Greenberg, A. (2015). New dark-web market is selling zero-day exploits to hackers. *Wired.* Retrieved from: http://www.wired.com/2015/04/therealdeal-zero-day-exploits/

Grunske, L., & Joyce, D. (2008). Quantitative risk-based security prediction for component-based systems with explicitly modeled attack profiles. *Journal of Systems and Software*, *81*(8), 1327-1345.

Heiberg, S., & Willemson, J. (2014). Modeling threats of a voting method.*Design, Development, and Use of Secure Electronic Voting Systems*, 128-148.

Henych, M. (2001). [Review of the book *Hackers: Crime in the Digital Sublime*, by Taylor, P.A.] *Policing: An International Journal of Police Strategies & Management* 24, no. 3 (2001): 432-434.

Hevner, A. R. (2004). Design Science In Information Systems Research. *MIS Quarterly*, *28*(1), 75-105.

Jürgenson, A., & Willemson, J. (2007). Processing multi-parameter attacktrees with estimated parameter values. In *Advances in Information and Computer Security* (pp. 308-319). Springer Berlin Heidelberg.

Jürgenson, A., & Willemson, J. (2008). Computing exact outcomes of multi-parameter attack trees. In *On the Move to Meaningful Internet Systems: OTM 2008* (pp. 1036-1051). Springer Berlin Heidelberg.

Jürgenson, A., & Willemson, J. (2010a). On fast and approximate attack tree computations. In *Information Security, Practice and Experience* (pp. 56-66). Springer Berlin Heidelberg.

Jürgenson, A., & Willemson, J. (2010b). Serial model for attack tree computations. In *Information, Security and Cryptology–ICISC 2009* (pp. 118-128). Springer Berlin Heidelberg.

Kjaerulff, U. B., & Madsen, A. L. (2008). Bayesian networks and influence diagrams. *Springer Science+ Business Media*, *200*, 114.

Kordy, B., Mauw, S., Radomirović, S., & Schweitzer, P. (2011). Foundations of attack–defense trees. In *Formal Aspects of Security and Trust* (pp. 80-95). Springer Berlin Heidelberg.

Kordy, B., Piètre-Cambacédès, L., & Schweitzer, P. (2014). DAG-based attack and defense modeling: Don't miss the forest for the attack trees. *Computer Science Review*, *13*, 1-38.

Kshetri, N. (2006). The simple economics of cybercrimes. *Security & Privacy, IEEE*, *4*(1), 33-39.

Lenin, A., Willemson, J., & Sari, D. P. (2014). Attacker profiling in quantitative security assessment based on attack trees. In *Secure IT Systems* (pp. 199-212). Springer International Publishing.

Lindenberg, S. (2001). Intrinsic motivation in a new light. *Kyklos*, *54*(2-3), 317-342.

Mauw, S., & Oostdijk, M. (2006). Foundations of attack trees. In *Information Security and Cryptology-ICISC 2005* (pp. 186-198). Springer Berlin Heidelberg.

McAfee (2014). Net Losses: Estimating the Global Cost of Cybercrime. Economic impact of cybercrime II. Center for Strategic and International Studies (June).

Niitsoo, M. (2010). Optimal adversary behavior for the serial model of financial attack trees. In *Advances in Information and Computer Security* (pp. 354-370). Springer Berlin Heidelberg.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research.*Journal of management information systems*, *24*(3), 45-77.

Pieters, W., & Davarynejad, M. (2015). Calculating adversarial risk from attack trees: Control strength and probabilistic attackers. In *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance* (pp. 201-215). Springer International Publishing.

Pieters, W., Hadžiosmanović, D., Lenin, A., Montoya Morales, A. L., & Willemson, J. (2014). TREsPASS: Plug-and-Play Attacker Profiles for Security Risk Analysis (Poster).

Poremba, S. (2015). The Internet Of Things Has A Growing Number Of Cyber Security Problems. *ForbesBrandVoice.* Retrieved from: http://www.forbes.com/sites/sungardas/2015/01/29/the-internet-of-things-has-a-growing-number-of-cyber-security-problems/

Rogers, M. K. (2006). A two-dimensional circumplex approach to the development of a hacker taxonomy. *Digital investigation*, *3*(2), 97-102.

Schneier, B. (1999). Attack trees. *Dr. Dobb's journal*, *24*(12), 21-29

Smith, A. D., & Rupp, W. T. (2002). Issues in cybersecurity; understanding the potential risks associated with hackers/crackers. *Information Management & Computer Security*, *10*(4), 178-183.

Torn, T. (2014). Security Analysis of Estonian I-Voting System Using Attack Tree Methodologies (Master's thesis, Tallinn University of Technology, Tallinn, Estonia). Retrieved from: http://digi.lib.ttu.ee/i/?1781

Turgeman-Goldschmidt, O. (2005). Hackers' accounts hacking as a social entertainment. *Social Science Computer Review*, *23*(1), 8-23.

Thycotic Software Ltd. (2014). *Thycotic Black Hat 2014 Hacker Survey Executive Report*. Retrieved from http://thycotic.com/wp-content/uploads/2014/03/Executive-summary-blackhat-survey-data-2014_PDF.pdf

TREsPASS. (2015). The TREsPASS project. *The TREsPASS project.* Retrieved from: http://www.trespass-project.eu/.

Van Kessel, P., & Allan, K. (2013). Under cyber attack. EY ' s Global Information Security Survey 2013, (October).

Van Kessel, P., & Allan, K. (2014). Get ahead of cybercrime. EY ' s Global Information Security Survey 2014, (October).

Weiss, J. D. (1991). A system security engineering process. In *Proceedings of the 14th National Computer Security Conference* (Vol. 249), 572-581.

# A Threat Agent Library

| | Intent | NON-HOSTILE | | | HOSTILE | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Employee Reckless | Employee Untrained | Info Partner | Anarchist | Civil Activist | Competitor | Corrupt Government Official | Data Miner | Employee Disgruntled | Government Cyberwarrior | Government Spy | Internal Spy | Irrational Individual | Legal Adversary | Mobster | Radical Activist | Sensationalist | Terrorist | Thief | Vandal | Vendor |
| Access (1) | Internal | ■ | ■ | ■ | | | | | | ■ | | ■ | ■ | | | | | | | ■ | | ■ |
| | External | | | | ■ | ■ | ■ | ■ | ■ | | ■ | | | ■ | ■ | ■ | ■ | ■ | ■ | | ■ | |
| Outcome (1-2) | Acquisition/Theft | | | | | | | | | | | | ■ | | | ■ | | | | ■ | | |
| | Business Advantage | | | | | | ■ | ■ | ■ | | | ■ | | | ■ | | | | | | | ■ |
| | Damage | ■ | ■ | ■ | ■ | | | | | ■ | ■ | | | ■ | | | ■ | ■ | ■ | | ■ | |
| | Embarrassment | ■ | ■ | ■ | | ■ | | | | ■ | ■ | | | ■ | ■ | | ■ | ■ | | | | |
| | Tech Advantage | | | | | | ■ | ■ | ■ | | | ■ | ■ | | | | | | | | | ■ |
| Limits (max) | Code of Conduct | | ■ | ■ | | | | | | | | | | | | | | | | | | |
| | Legal | ■ | | | | | | | | | | | | | ■ | | | | | | | ■ |
| | Extra-legal, minor | | | | | ■ | ■ | ■ | ■ | | | | ■ | | | | ■ | ■ | | ■ | ■ | |
| | Extra-legal, major | | | | ■ | | | | | ■ | ■ | ■ | | ■ | | ■ | | | ■ | | | |
| Resources (max) | Individual | ■ | ■ | ■ | | | | | | ■ | | | | ■ | | | | | | ■ | | |
| | Club | | | | ■ | | | | | | | | | | | | | ■ | | | | |
| | Contest | | | | | | | | | | | | | | | | | | | | ■ | |
| | Team | | | | | | | | ■ | | | | | | | | | | | | | ■ |
| | Organization | | | | | ■ | ■ | | | | | | ■ | | ■ | ■ | ■ | | ■ | | | |
| | Government | | | | | | | ■ | | | ■ | ■ | | | | | | | | | | |
| Skills (max) | None | | | | ■ | | | | | | | | | ■ | | | | | | ■ | | |
| | Minimal | | ■ | | | | | | | | | | | | | | | ■ | | | | |
| | Operational | | | ■ | | | | | | ■ | | | | | | | | | | | ■ | ■ |
| | Adept | ■ | | | | ■ | ■ | ■ | ■ | | ■ | ■ | ■ | | ■ | ■ | ■ | | ■ | | | |
| Objective (1 or more) | Copy | | | | | ■ | ■ | | ■ | | | ■ | ■ | | ■ | | | | | | | ■ |
| | Deny | | | | | | | ■ | | | ■ | | | | | | | | | | | |
| | Destroy | | | | ■ | | | | | ■ | ■ | | | | | | | | ■ | | | |
| | Damage | | | | | | | | | ■ | ■ | | | | | | | | ■ | | | |
| | Take | | | | | | | | | | | | | | ■ | ■ | | | ■ | ■ | | |
| | All of the Above/ Don't Care | ■ | ■ | ■ | | | | | | | | | | ■ | | | ■ | ■ | | | ■ | |
| Visibility (min) | Overt | | ■ | | ■ | | | ■ | | | | | | | ■ | | ■ | ■ | | | | |
| | Covert | ■ | | | | ■ | | | | | | | | | | ■ | | | ■ | | ■ | |
| | Clandestine | | | ■ | | | ■ | | ■ | | | ■ | ■ | | | | | | | ■ | | ■ |
| | Multiple/Don't Care | | | | | | | | | ■ | ■ | | | ■ | | | | | | | | |

*Figure 18: Intel Threat Agent Library (Casey, 2007)*

# B Attack tree tree on I-voting

| Identifier | Node | Type |
|---|---|---|
| 1 | Manipulation attack | Root |
| 1.1 | Attack voters' environment | OR |
| 1.1.1 | Malware | OR |
| 1.1.1.1 | Vote modifying malware | AND |
| 1.1.1.1.1 | Develop malware | OR |
| 1.1.1.1.1.1 | Vote changing malware | Leaf |
| 1.1.1.1.1.2 | Vote blocking malware | Leaf |
| 1.1.1.1.2 | Distribute malware | AND |
| 1.1.1.1.2.1 | Compromise voters'computers | OR |
| 1.1.1.1.2.1.1 | Create botnet | Leaf |
| 1.1.1.1.2.1.2 | Buy botnet | Leaf |
| 1.1.1.1.2.2 | Compromise voters' mobile devices | OR |
| 1.1.1.1.2.2.1 | Create mobile botnet | Leaf |
| 1.1.1.1.2.2.2 | Buy m0bile botnet | Leaf |
| 1.1.1.1.3 | Avoid detection | Leaf |
| 1.1.1.2 | Re-voting malware | AND |
| 1.1.1.2.1 | Develop malware | Leaf |
| 1.1.1.2.2 | Compromise voters' computers | Or |
| 1.1.1.2.2.1 | Create botnet | Leaf |
| 1.1.1.2.2.2 | Buy botnet | Leaf |
| 1.1.1.2.3 | Avoid detection | Leaf |
| 1.1.1.3 | Self-voting malware | AND |
| 1.1.1.3.1 | Develop malware | Leaf |
| 1.1.1.3.2 | Compromise voter's computer | OR |
| 1.1.1.3.2.1 | Create botnet | Leaf |
| 1.1.1.3.2.2 | Buy botnet | Leaf |
| 1.1.1.3.3 | Avoid detection | Leaf |
| 1.1.2 | Fake voting applications | AND |
| 1.1.2.1 | Develop fake apps | AND |
| 1.1.2.1.1 | Develop fake Voting App. | Leaf |
| 1.1.2.1.2 | Develop fake Verification App. | Leaf |
| 1.1.2.2 | Distribute fake Voting App. | OR |
| 1.1.2.2.1 | Use fake website | AND |
| 1.1.2.2.1.1 | Develop fake website | LEAF |
| 1.1.2.2.1.2 | Get voters to visit fake website | OR |
| 1.1.2.2.1.2.1 | E-mail | Leaf |
| 1.1.2.2.1.2.2 | Network attacks | Leaf |
| 1.1.2.2.1.2.3 | Social media | Leaf |
| 1.1.2.2.2 | Replace app. on NEC web server | OR |
| 1.1.2.2.2.1 | Bribe server admin | Leaf |
| 1.1.2.2.2.2 | Bribe SW developer | Leaf |
| 1.1.2.2.2.3 | Exploit configuration error | Leaf |
| 1.1.2.3 | Distribute fake Verification App. | OR |
| 1.1.2.3.1 | From official appstore | OR |
| 1.1.2.3.1.1 | Upload similar | Leaf |
| 1.1.2.3.1.2 | Replace original | OR |
| 1.1.2.3.1.2.1 | Bribe server admin | Leaf |
| 1.1.2.3.1.2.2 | Bribe SW developer | Leaf |
| 1.1.2.3.1.2.3 | Exploit configuration error | Leaf |
| 1.1.2.3.2 | From other markets | Leaf |
| 1.1.2.4 | Avoid detection | Leaf |
| 1.2 | Attack Central System | OR |
| 1.2.1 | Compromise VSS | AND |
| 1.2.1.1 | Develop malicious code | Leaf |
| 1.2.1.2 | Insert code into server | OR |
| 1.2.1.2.1 | Bribe server admin | Leaf |
| 1.2.1.2.2 | Bribe SW developer | Leaf |
| 1.2.1.2.3 | Get access to server | AND |
| 1.2.1.2.3.1 | Get access to internal network | Leaf |
| 1.2.1.2.3.2 | Exploit configuration error | Leaf |
| 1.2.2 | Compromise VCA | AND |
| 1.2.2.1 | Develop malicious code | Leaf |
| 1.2.2.2 | Insert code into server | OR |
| 1.2.2.2.1 | Bribe server admin | Leaf |
| 1.2.2.2.2 | Bribe SW developer | Leaf |
| 1.2.3 | Compromise data carrier | AND |
| 1.2.3.1 | Get access to device | OR |
| 1.2.3.1.2.1 | Bribe worker | Leaf |
| 1.2.3.1.2.1 | Infiltrate as participant | Leaf |
| 1.2.3.2 | Compromise device | Leaf |