A framework for the motivation of attackers in attack tree analysis

Rick van Holsteijn, Wolter Pieters, Maarten Franssen, Jan van den Berg, Pieter van Gelder

Delft University of Technology

Abstract. Cyberattacks are becoming more and more frequent, which raises the need for good cybersecurity practices. IT security experts are tied to budgets, which is why methodologies are developed to help them allocate resources. The attack tree is one of these methodologies. In current methods there is no possibility to take into account the motivation of the attacker. By means of a design science approach a framework has been designed that includes the motivation of attackers in the attack tree analysis. The designed framework provides more flexibility in the pay-off values for attackers. With the use of the framework, it is possible to differentiate the pay-off values for variously motivated attackers as well as for various attack paths.

Keywords: attack tree, motivation, attacker profile, pay-off, framework

1 Introduction

In recent years, cyberattacks have become more and more complex and the amount of attacks is expected to keep multiplying in the coming years [1,2]. This stresses the need for good cybersecurity practices, but most IT security experts are bound by budgets and they often do not have a very good insight in the threats that exist [2]. In order to help these experts get a clearer overview of the threats on their system and to help them make decisions on what countermeasures to take, various tools have been developed.

One of the tools developed for cyber security experts is the attack tree methodology. With this methodology it is possible to analyze complex attacks that consist of multiple steps and where multiple steps are possible [3]. By splitting up the overall attack into smaller steps, it is possible to derive the security properties from the properties of the smaller steps [4]. After the introduction of the method by Schneier [5], the methodology has been developed further over time [3,4], [6-15]. A full overview of all the various ways in which the methodology is described is found in [16].

One way in which the attack tree methodology has been developed is by making the parameters that are assigned to the attack tree independent of the type of attacker [12,13]. In this way the attack tree can be reused for various types of attackers, without having to update all the parameter values. The attacker properties are in this case separated from the system properties and are represented in attacker profiles.

Various studies have been performed to form attacker profiles [17-22]. One of the attacker characteristics that was mentioned in almost all of them was the attacker's motivation. Within the current attack tree methodology, this attacker characteristic has not yet been separated from the parameter values. The framework that was designed in the current research aims at resolving this research gap, by including the motivation of attackers in the attack tree methodology. The research question related to this research is the following: *How can the motivation of attackers be included in the use of attack trees for cyber threat analysis?* A design science research approach described by [23] was used to form the framework.

This paper describes the framework and the most important steps taken during the design process. In section 2 the state of the art in the attack tree methodology is described. Section 3 describes the value that the framework adds and section 4 describes the actual design process and presents the framework. In section 5 an example will be worked out, to show how to apply the framework. Some concluding notes are given in section 6.

2 The attack tree methodology

The first use of a tree structure for analyzing vulnerabilities of a system was presented by Weiss [24]. The term attack tree was however introduced a few years later by Schneier [5] and he is therefore often stated as the pioneer in the field. The attack tree methodology introduced by Schneier consists of two phases. First the attack tree is constructed. To construct the attack tree first an overall goal for the attacker is chosen. This goal is represented in the root node of the attack tree. This root node is then split into sub attacks, or sub goals. You continue this process until you cannot split them any further and these attacks should be of such an elementary level that it is possible to assign parameters to them. Each node that is split up can either be 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.

After constructing the attack tree comes the analysis phase. In this phase you assign parameter values to the attack tree and analyze the results. Schneier mentions various possibilities for parameter values, but the most basic one is a Boolean parameter that takes either the value 'possible' or 'impossible' [5]. What can be seen from this, is that Schneier has to assume a certain type of attacker when assigning the parameter values. Whether an attack is possible or impossible depends on the type of attacker. The parameter is thus not attacker independent, which would be needed for reusability of the attack tree.

In early uses of the attack tree methodology, just a single parameter was assigned. Later on a model was developed that uses multi-parameter attack trees [6]. This was further developed by Jürgenson & Willemson [8,9] and in [14] an effort was made to make various parameters attacker independent, by analyzing attacker properties separately. In the method of [14] the attacker properties skill and resources were taken into account. The current framework has extended this method of [14] by also taking into account the motivation of the attacker.

3 Added value of the framework

In order to include the motivation within the framework for the attack tree methodology, the influence of the attacker's motivation has to be clear. The motivation of the attacker relates to the reason for performing the attack. If two differently motivated attackers will perform the same attack they will probably value the outcome differently. If we assume an attacker that is motivated by financial benefits performs a challenging attack that yields no money he will value the outcome very low. For an attacker that is motivated by the challenge the attack provides, the outcome of the same attack may be valued a lot higher. The motivation of the attacker thus influences the pay-off of the attack for an attacker.

In the current model this pay-off is represented by a global gains parameter, that has the same value for each type of attacker [7-10], [12]. In this case you would thus have to update the gains parameter when assuming different types of attackers. Also the gains parameter has low flexibility in the current model, because every path an attacker takes, leads to the same gains. It is however likely that 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. These shortcomings of the current methodology are overcome by the designed framework. In the new model:

- 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

4 Description of the framework

The framework designed consists of three different parts and is visualized in Fig.1. The first part is to set up the attack tree, the second to set up the attacker profile and the third part is to combine the two and analyze the results. These three parts are discussed in consecutive subsections.

4.1 Setting up the attack tree

The first step in setting up the attack tree, is to determine the main objective of the attacker, which will be the root node of the attack tree. After doing so, the rest of the tree can be constructed by splitting up the root node and keep splitting the nodes until no further refinement is necessary.

The next step is to assign parameter values to each of the leaf nodes. The parameters that are used within the framework are the same as those in the method of Lenin et al. [12]. A value for the following parameters needs to be assigned:

- Expenses – This parameter consists of the costs for the attack and the possible penalties when being caught. Values for this parameter are real numbers.
- Difficulty – This parameter describes the difficulty of the attack and is chosen on an ordinal scale. Possible values are {*low*, *medium*, *high*, *very high*}.
- Required attack time – This parameter describes the time that is required to perform the attack. The parameter is measured on an ordinal scale and the possible values are {*seconds*, *minutes*, *hours*, *days*}.
- Probability of success – This parameter describes the chance that the attack is successfully performed if attempted. This parameter has a value between 0 and 1.

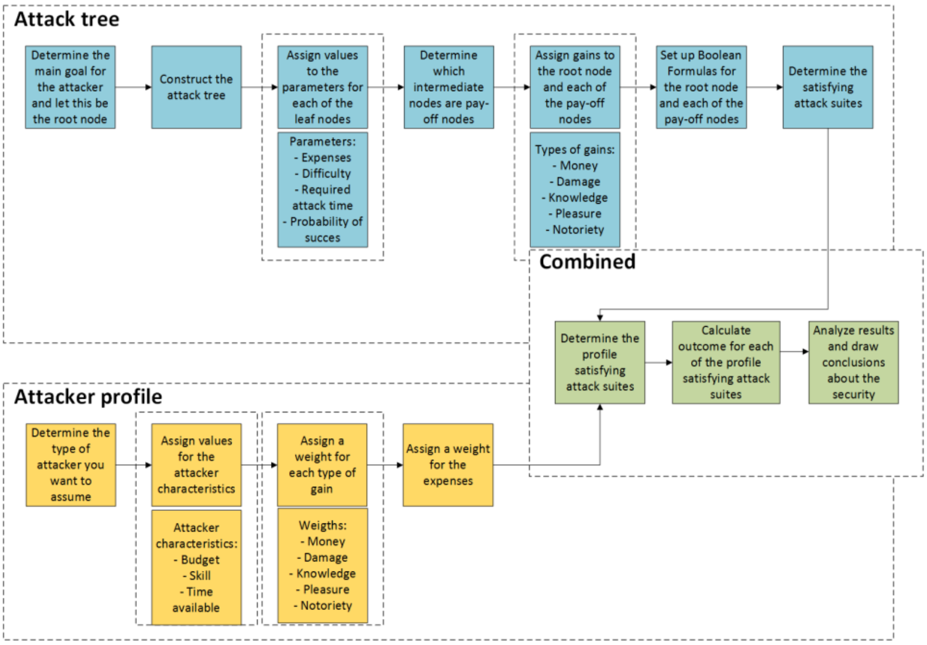
Attack tree
Determine the main goal for the attacker and let this be the root node
Construct the attack tree
Assign values to the parameters for each of the leaf nodes
Parameters: - Expenses - Difficulty - Required attack time - Probability of succes
Determine which intermediate nodes are pay-off nodes
Assign gains to the root node and each of the pay-off nodes
Types of gains: - Money - Damage - Knowledge - Pleasure - Notoriety
Set up Boolean Formulas for the root node and each of the pay-off nodes
Determine the satisfying attack suites
Combined
Determine the profile satisfying attack suites
Calculate outcome for each of the profile satisfying attack suites
Analyze results and draw conclusions about the security
Attacker profile
Determine the type of attacker you want to assume
Assign values for the attacker characteristics
Attacker characteristics: - Budget - Skill - Time available
Assign a weight for each type of gain
Weigths: - Money - Damage - Knowledge - Pleasure - Notoriety
Assign a weight for the expenses

Fig. 1. Visualization of the framework for the attack tree methodology

After assigning the parameters, intermediate pay-off nodes need to be identified, which are the nodes that, when reached, result in gains for the attacker. The inclusion of intermediate pay-offs makes it possible to have different gains for different ways of performing the attack.

The next step is to assign a value for the gains for the root node as well as for the intermediate pay-off nodes. In the new framework, these gains are no longer a single value, but are split up in five different forms of gains. Splitting up the gains allows you to differentiate between gains for different types of motivation. For an attacker motivated by financial benefits, money is the most important gain, but for an attacker motivated by notoriety, fame is the most important gain. A list has been formed of the possible motivations an attacker may have. This list is based on previous studies on the motivation of attackers [17-22], [25,26]. For each of these motivations, an associated gain has been identified. Table 1. shows the types of motivation and their associated gains.

Table 1. Motivations and their associated gains

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

When assigning values for each of these types of gains, it is important to assign values that are independent of the attacker. This is to make sure that the attack tree can be analyzed for variously motivated attackers, without having to update the parameters in the attack tree. The attacker profile that contains the attacker characteristics is used to differentiate between pay-offs for various attackers. The average pay-off value for an attacker is suggested to be used as the attacker independent gain value.

Estimating values for each of the types of gains is a complex task, as it is hard to quantify pleasure for example. Also each of the gains needs to be measured on the same scale as to be able to compare them with each other. The recommended way to do so is by monetizing the values for each type of gain. This way all the gains will be expressed in money and will thus be comparable. Estimating monetary values for the gains is however not straightforward. One way of doing so, would be by using guesstimates, where the IT security expert him/herself attempts to estimate the gains. The expert could for example use available sources on previous incidents or his/her experience to make a plausible estimate.

Inspiration for estimating the gain values can also be taken from the way in which insurance companies put prices on injuries or deaths. The two ways that are used to do so are based human capital and on willingness to pay [27]. Using these methods is however very time consuming and may therefore not be useful. A combination of the methods could also be used, to be able to choose the most applicable method for each of the types of gains.

The next step is to set up a Boolean formula for the whole tree and for every sub tree rooted in a pay-off node. In the current methodology, the attack tree is represented by a Boolean formula where each elementary attack is a Boolean value that is set to True if the elementary attack is attempted and False if the elementary attack is not attempted. The elementary attacks that will be attempted are represented in an attack suite. The overall Boolean formula returns True if the root node is reached by the attack suite and False, if not. How this Boolean formula takes form, will be shown in the example in chapter 5. In the current methodology, only attack suites that return a True value in the Boolean formula are taken into account in the analysis. In the designed framework there are however also intermediate pay-off nodes and we want to allow the attacker an opt-out possibility as well. This means that also attack suites that just reach an intermediate pay-off node should be taken into account in the analysis. In order to do so, not just one Boolean formula is formed, but a set of formulas. One for the overall tree and one for every subtree that is rooted in a pay-off node.

The last step in setting up the attack tree is to determine what the satisfying attack suites are. A satisfying attack suite within the designed framework is an attack suite that returns True for at least one of the set of Boolean formulas.

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. The next subsection discusses the setting up of the attacker profile.

4.2 Setting up the attacker profile

The next part of the framework focusses on setting up the attacker profile. The first step in setting up the attacker profile is to determine what type of attacker you want to analyze. For this type of attacker, values need to be assigned for various attacker characteristics. The characteristics for which a value needs to be assigned are the following:

- Budget – This parameter describes the amount of money that an attacker has available. The value for this parameter is a real number.
- Skill – This parameter describes the capabilities of the attacker and is measured on an ordinal scale. The possible values are {*low, medium, high*}.
- Time available – This parameter describes the time that the attacker has available for performing an attack. Values are on an ordinal scale, which are {*seconds, minutes, hours, 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 discussed earlier, the attacker independent gain value is considered to be the average pay-off value for an attacker. The weight value is used to form the pay-off value for the attacker described by the attacker profile. The weight value should say something about how the attacker values the gain as compared to the average attacker. Two possible ways of allocating these weight values are:

1. Free choice of weight values
2. Choose weight from predefined set of values

The first method give the IT security expert the most freedom in choosing the weight values, which may be necessary is s/he is interesting in calculating the exact outcome values for a certain attacker. However even with this freedom it may still not be possible to calculate the actual pay-offs, because this is a very complex task.

It is more realistic for the IT security expert to form a ranking of the outcomes of the various attacks. In this case no exact pay-off values are necessary and thus the second method for allocating weight values can be used. For example only values between zero and two could be chosen. In this case it is possible to indicate that the attacker in not interested in the type of gain at all or that he is interested twice as much as the average attacker.

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 can possibly form an outcome here.

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. Also in the case of assigning weight values between 0 and 1 to the various types of gains, the expenses could get too much influence if no weight is assigned to these expenses. A weight will therefore also be assigned to the expenses.

This concludes the steps needed to set up the attacker profile. The result is a profile of a certain attacker with his/her characteristics and weights for the various types of gains and the expenses. The next sub section describes how the attack tree and attacker profile are combined.

4.3 Combining the attack tree and the attacker profile

After setting up the attack tree and the attacker profile, the two can be combined and the actual analysis can be performed. The first step here is to determine which attack suites are profile satisfying. An attack suite is profile satisfying if an attacker is able to perform the attacks within the attack suite. There are three ways in which an attack suite can fail to be attacker satisfying [12]:

- If the total expenses of the elementary attacks in the attack suite are higher than the budget of the attacker.
- If any of the elementary attacks in the attack suite has a higher difficulty level than the skill level of the attacker.
- If any of the elementary attacks in the attack suite has a higher required attack time than the time available of the attacker.

The next step is to follow the mathematical structure to calculate the outcome for each of the profile satisfying attack suites. In the analysis the attackers are assumed to be rational. The goal of the analysis is thus to find the attack suites that have positive outcome, because when an attacker performs these attack suites there is a positive expected result. The outcome of an attack suite is calculated by subtracting the expenses from the pay-offs. The formula used for this is the following:

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

In this formula, the sigma represents the attack suites which is a set elementary attacks X_i. The pay-off and thus the outcome is dependent of the attacker *j*. The weight for the expenses as defined in the attacker profile is represented by e^j. The *Expenses* are formulated in the attack tree.

The pay-off of an attack suite is calculated by summing up all the pay-offs of every pay-off node that is reached multiplied by the probability of successfully performing the attack in that pay-off node. In formula form this looks as follows:

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

In this formula $F\big(T(Y_i)\big)$ represents the Boolean formula for the (sub) tree with its rood in node Y_i. Both the intermediate nodes and the root node are represented by Y_i, where Y_0 is the root node. The pay-off for attacker *j* in pay-off node Y_i is represented by P_i^j. The last part of the formula is $p_{\sigma,T(Y_i)}$ which is the probability of success for the (sub) tree rooted in Y_i for attack suite sigma. The probability of success is calculated up the attack tree by using the following process:

- For each of the elementary attacks that is not in sigma the probability of success is set to 0
- For each of the elementary attacks in the attack suite 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)

Following this process for every node in the attack tree, you eventually reach the pay-off node for which you need to determine the probability of success. This is the probability of success you need in the formula.

The last formula that is needed is for calculating the pay-off for a certain attacker. This formula has the form of a utility function where the value for each type of gain in the attack tree is multiplied by its associated weight value. The formula looks as follows:

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

In this formula w_k^j represents the weight of attacker j for gain type k and g_k^i represents the gain of pay-off node Y_i for gain type k.
The last step of the framework is to analyze the results and draw conclusions on the security of the system. Based on the outcomes the IT security expert can decide whether or not to take countermeasures and where to place them.

5 Example application of the framework

To further clarify the described framework an example is presented. The example deals with an attack tree in which the main goal is to obtain secret data. The way to do so is by stealing a laptop AND decrypting the data. For stealing a laptop the attacker 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 is visualized in Fig.2. The link under the 'Obtain secret data' node indicates that it is an AND node.

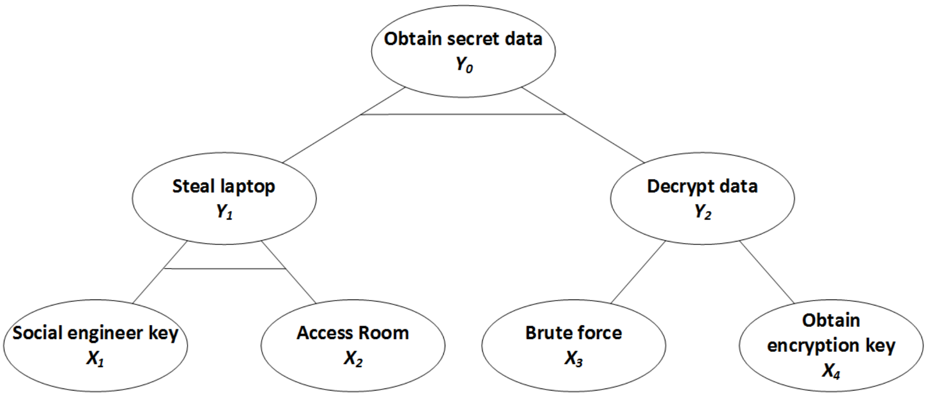

Fig. 2. Attack tree for obtaining secret data

After setting up the attack tree, parameter values are assigned to the elementary attacks. The parameter values assigned are shown in Table 2.

Table 2. Parameter values for the 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 3.

Table 3. Gains values for each 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 of 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 $\{X_1, X_2, X_3, X_4\}$.

The next part is to set up the attacker profile. In order to show the influence of the weight values, two attacker profiles are set up. The values for the attacker characteristics that are assigned, are the same for both attackers and shown in Table 4.

Table 4. Values for the attacker characteristics

Attacker characteristic	Profile value
Budget	1000
Skill	High
Time available	Days

The last two steps of setting up the attacker profile are assigning the weights for the various types of gains and assigning the weight for the expenses. These weights are combined and shown in Table 5.

Table 5. 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

For the analysis phase, the first step is to determine which attack suites are profile satisfying. In this example all of the attack suites are profile satisfying and the outcome will thus be calculated for the attack suites $\{X_1, X_2\}$, $\{X_1, X_2, X_3\}$, $\{X_1, X_2, X_4\}$ and $\{X_1, X_2, X_3, X_4\}$. The previously presented formulas are used for this and the results for attacker 1 are presented in Table 6 and for attacker 2 in Table 7.

Table 6. Satisfying attack suites and the calculated outcome for attacker 1

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

Table 7. Satisfying attack suites and the calculated outcome for attacker 2

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

In the case of this example, you could say 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.

6 Concluding notes

This paper describes a framework that was designed to resolve the research gap associated with the research question: *How can the motivation of attackers be included in the use of attack trees for cyber threat analysis?* An example was given to demonstrate the designed framework. The framework has also been tested on a real world case, where the framework was considered to improve the current state of the art. The framework allows for various pay-offs for variously motivated attackers and more flexibility is possible in the gains parameter, because of the inclusion of intermediate pay-offs and because the gains parameter is split in different forms of gains.

There is however still room for improvement. The main shortcoming of the framework can be found in the guidelines for estimating the gains. No clear description has yet been given on how these gains can be estimated independent of the attacker. Future research could look into the possibility of changing these gains in interval values, where no exact value needs to be given. The method described in [7] could possibly serve as a starting point for that. Also looking into the possibility of including a sensitivity analysis for the outcomes might prove fruitful in the future.

Also for setting the weights, no definitive method was provided. Some possibilities are described with their pros and cons, but further research will be necessary to determine the most suitable method.

References

1. Van Kessel, P., & Allan, K. (2013). Under cyber attack. EY ' s Global Information Security Survey 2013, (October).
2. Van Kessel, P., & Allan, K. (2014). Get ahead of cybercrime. EY ' s Global Information Security Survey 2014, (October).
3. 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.
4. Mauw, S., & Oostdijk, M. (2006). Foundations of attack trees. In *Information Security and Cryptology-ICISC 2005* (pp. 186-198). Springer Berlin Heidelberg
5. Schneier, B. (1999). Attack trees. *Dr. Dobb's journal*, *24*(12), 21-29
6. 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.
7. 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.
8. 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.
9. Jürgenson, A., & Willemson, J. (2010). On fast and approximate attack tree computations. In *Information Security, Practice and Experience* (pp. 56-66). Springer Berlin Heidelberg.
10. Jürgenson, A., & Willemson, J. (2010). Serial model for attack tree computations. In *Information, Security and Cryptology–ICISC 2009* (pp. 118-128). Springer Berlin Heidelberg.
11. 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.

12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. Barber, R. (2001). Hackers profiled—who are they and what are their motivations?. *Computer Fraud & Security*, *2001*(2), 14-17.
18. Rogers, M. K. (2006). A two-dimensional circumplex approach to the development of a hacker taxonomy. *Digital investigation*, *3*(2), 97-102.
19. Casey, T. (2007). Threat Agent Library Helps Identify Information Security Risks. *Intel White Paper, September*.
20. Casey, T. (2015). Understaning Cyberthreat Motviations to Improve Defense. *Intel White Paper.*
21. Turgeman-Goldschmidt, O. (2005). Hackers' accounts hacking as a social entertainment. *Social Science Computer Review*, *23*(1), 8-23.
22. 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.
23. 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.
24. Weiss, J. D. (1991). A system security engineering process. In *Proceedings of the 14th National Computer Security Conference* (Vol. 249), 572-581.
25. Kshetri, N. (2006). The simple economics of cybercrimes. *Security & Privacy, IEEE*, *4*(1), 33-39
26. 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.
27. Etter, I. B. (1987). The National Safety Council's estimates of injury costs. *Public Health Reports (1974-)*, 634-636.