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

Attacker Profiling in Quantitative Security Assessment
Master’s Thesis

Student: Dyan Permata Sari
Student code: 106967IVCMM
Supervisor: Prof. Ahto Buldas, Ph.D
Co-Supervisor: Aleksandr Lenin, M.Sc

Tallinn
2014
Declaration

I hereby declare that I am the sole author of this thesis. The work is original and has not been submitted for any degree or diploma at any other University. I further declare that material obtained from other sources has been dully acknowledged in the thesis.

................................. .................................
(Date) (Author’s signature)
## List of Abbreviations and Symbols

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP</td>
<td>Border Gateway routing Protocol</td>
</tr>
<tr>
<td>SERVE</td>
<td>Secure Electronic Registration and Voting Experiment</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identification Systems</td>
</tr>
<tr>
<td>TAN</td>
<td>Transaction Authentication Number</td>
</tr>
<tr>
<td>CS</td>
<td>Customer Service</td>
</tr>
<tr>
<td>SCADA</td>
<td>Industrial process control systems</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Attack tree $T$ with AND-nodes, OR-nodes and set of leaves $X = {X_1, \ldots, X_n}$ and parameter Gains</td>
</tr>
<tr>
<td>$F$</td>
<td>Boolean formula $F$ defined by the syntax tree of the attack tree $T$</td>
</tr>
<tr>
<td>$X_i$</td>
<td>Leaf (elementary attack) $X_i$ of the attack tree $T$ with parameters $P_i$ and Expenses$_i$</td>
</tr>
<tr>
<td>Expenses$_i$</td>
<td>Expected cost of launching the elementary attack $X_i$, which includes the preparation costs and expected penalties</td>
</tr>
<tr>
<td>$P_i$</td>
<td>Probability of succeeding with the elementary attack $X_i$</td>
</tr>
<tr>
<td>Gains</td>
<td>Reward of an attacker, when the attack tree $T$ is realized by the attack suite $S$</td>
</tr>
<tr>
<td>$S$</td>
<td>Attack suite $S \subseteq X$ of the elementary attacks in the attack tree $T$</td>
</tr>
<tr>
<td>Outcome$_S$</td>
<td>Outcome value of the attack suite $S$</td>
</tr>
<tr>
<td>$P_S$</td>
<td>Probability of $F$ = true after executing the attack suite $S$</td>
</tr>
</tbody>
</table>
Abstract

In the past decades of cyber warfare, cyber-attack has grown its complexity in attack patterns and impacted a vast geographical area, therefore attack trees are suitable to describe possible security threat of a system. An attack tree consists of root node that represents attacker’s goal and leaf nodes constituting as attack steps required to be executed by an attacker to achieve the successful state. Development of an attack tree for a system may begin with a simple tree, but later will allow to refine these attacks into more atomic steps. There are two types of refinements which commonly used nowadays – the conjunctive refinement and the disjunctive one.

Quantitative analysis based on attack trees, utilizes bottom-up parameter propagation approach. Single-parameter propagation method was introduced in 1999 by Schneier [1], later on improved by Mauw et al. [2]. This method assigns single quantitative parameter for each leaf in an attack tree and propagates this parameter using bottom-up technique towards the root node. Furthermore, the final value of the root node as the result of the analysis is used to form conclusions. However, another method was introduced back in 2006 which suggests to assign many parameters for each leaf in an attack tree, to calculate attacker’s utilities in the leaf nodes and propagate it towards the root node. Buldas et al. called it multi-parameter approach [3].

We propose a new factor to be considered for the attack tree based quantitative analysis. “Attacker profile” is a formalized description of malicious actor’s resources and capabilities or skill levels. Reason for this proposal comes from difficulties in estimating parameters of leaves in an attack tree, which values depend on available resources and set of skills of malicious actors.

In addition, we would like to highlight that application of “Attacker Profile” to an attack tree can invalidates some nodes and thus will eliminate several sub-trees of the overall attack tree. Therefore, the resulting tree describes possible attack steps of the particular attacker. Furthermore, we investigate and analyze if attacker profile is a useful concept for quantitative assessment of security based on attack tree methodology.
# Table of Contents

1. Introduction ........................................................................................................ 9
   1.1 Motivation ........................................................................................................ 10
   1.2 Scope ............................................................................................................. 10
   1.3 Aim ............................................................................................................... 11
   1.4 Summary .................................................................................................... 11
   1.5 Outline ....................................................................................................... 11

2. Introduction to Security Modeling ................................................................ 12
   2.1 Concept of Security Modeling ................................................................. 12
   2.2 Attack Tree Analysis ................................................................................ 14
      2.2.1 Attack Graph ...................................................................................... 16
      2.2.2 Attack Tree Application .................................................................... 17
   2.3 Attack Tree Attributes ............................................................................. 17
      2.3.1 Concept of Attack Tree Attributes .................................................. 17
      2.3.2 Quantitative Analysis Calculations ............................................... 19
   2.4 Boolean Formula .................................................................................... 19

3. Related Work .................................................................................................. 20
   3.1 Attacker Profile Overview ..................................................................... 20
   3.2 Jürgenson - Willemson Parallel Model Logic .................................... 21
   3.3 Attacker Profile Parameters .................................................................. 22
   3.4 Attacker Profile Formal Definition ....................................................... 23
   3.5 Attacker Profile and Parallel Model of Multi-Parameter Attack Tree .. 24
      3.5.1 ApproxTree Calculation Method .................................................... 25

   4.1 Attack Tree Model ................................................................................... 26
      4.1.1 Attacker Profile: Types of Actors Involved .................................... 32
   4.2 Attack Tree Decoration with Attributes ............................................. 33

5. Observation and Discussion ...................................................................... 36
   5.1 ApproxTree Calculation Result ............................................................. 37
   5.2 Analysis .................................................................................................. 45
      5.2.1 Attack Tree and Attacker Profile Attribute Decoration ............ 45
5.2.2 Calculation Result Analysis ................................................................. 47
6. Conclusions and Future Research ............................................................ 49
Résumée ........................................................................................................ 52
References ..................................................................................................... 53
Appendix 1 .................................................................................................... 62
Appendix 2 .................................................................................................... 63
List of Figures

Figure 1. Example of a threat logic tree, originating from Weiss [10] paper: Unix System Design

Figure 2. Example simple Attack Tree assigned with value I = Impossible; P = Possible, Schneier [1]

Figure 3. Fraudulent Purchases of Digital Goods attack tree

Figure 4. Child Node attack tree “Account Take Over”

Figure 5. Refinement child node “Usage Policy Violation”

Figure 6. Refinement child node “CS Agents compromised Account”

Figure 7. Refinement “Infect Victim PC with Malware”

Figure 8. Refinement “Shoulder Surfing”

Figure 9. Refinement “Sniffing Network Traffic”

Figure 10. Attack Path based on calculation result for Attacker profile 1,3,4,5

Figure 11. Attack Path based on calculation result for Attacker profile 2

Figure 12. Attack Path based on calculation result for Attacker profile 6

Figure 13. Attack Path based on calculation result for Attacker profile 7

Figure 14. Attack Path based on calculation result for Attacker profile 8

Figure 15. Attack Path based on calculation result for Attacker profile 14

Figure 16. Attack Path based on calculation result for Attacker profile 16,17,18

Figure 17. Attack Path based on calculation result for Attacker profile 19

Figure 18. Attacker profile invalidate nodes and form an appropriate Attack Suites
List of tables

Table 1. Attacker Profile Sample for “Fraudulent Purchases of Digital Goods”


Table 2. Decoration attributes for “Fraudulent Purchases of Digital Goods” Attack Tree

Table 3. Estimation Parameter value for child nodes

“Fraudulent Purchases of Digital Goods” Attack Tree

Table 4. Overall calculation Result

Table 5. Calculation result for Attacker Profile 1,3,4,5

Table 6. Calculation result for Attacker Profile 2

Table 7. Calculation result for Attacker Profile 6

Table 8. Calculation result for Attacker Profile 7

Table 9. Calculation result for Attacker Profile 7

Table 10. Calculation result for Attacker Profile 10,11,13,14

Table 11. Calculation result for Attacker Profile 16,17,18

Table 12. Calculation result for Attacker Profile 19
Chapter 1
Introduction

As threats in cyber world have become more complicated and sophisticated, it is necessary to provide comprehensive and formalized method to evaluate systems security. Schneier in 1999 [1] introduced a suitable method which can be used to analyze possible security threats or weaknesses of a system. Studies on improving attack trees as the basis of security assessments has been done by many researchers and the development itself has delivered newer methods which can accommodate the complexity of security threats toward modern systems.

Quantitative assessment of attacks has become an active research subject. First quantitative method that presented by Schneier in 1999. The idea is to propose attacker’s ultimate goal as root node and refine it into sub-goals. There are two types of refinements that are used nowadays, the conjunctive refinement, in case of which all sub-goals must be satisfied to satisfy the goal and the disjunctive refinement, in case of which any of the sub-goals is sufficient to satisfy the goal. Moreover, the non-refined nodes or leaves of attack tree describe attack steps that are performed by an attacker. In addition to the refinement approach, Schneier [1] also assigned single parameter to each leaf in an attack tree and propagated initial parameter towards the root node.

Buldas et al. [3] in 2006, introduced multi-parameter attack trees which highlight the attribute called expected attacker’s outcome. Attacker’s outcome represents of monetary gain of the attacker and is determined by several parameters: gains of the attacker in case the attack succeeds, costs of the attack, success probability of the attack, probability of getting caught and expected penalties in case of being caught. An attack tree model by Buldas et al. [3] is not consistent with Mauw et al. [2] foundations, equivalent attack trees should result in the same outcome values. Therefore, Jürgenson et al. [4] presents new a concept for multi-parameter approach called as Parallel Attack Tree Model, which is consistent with Mauw et al. model [2] and gives outstanding outcome value compared to initial multi-parameter attack tree approach of Buldas et al. [3].

Multi-parameter attack tree computations assume rational behavior of an attacker. This means that elementary attacks, carried out by the attacker will unlikely have greater cost than the final gain. Thus attacker chooses the most profitable way to attack. We will introduce attacker profile as a
separate parameters in an attacker tree. We will estimate the quantitative parameters for leaves of the tree in an actor-independent way as much as possible, for instance brute-force password hash is an easy attack for medium skilled and experienced attacker but might be difficult for unskilled attacker who are only rely on ready to use dictionary attack tool. Another example is social-engineering attacks, the value of sub-goal to compromise or take-over credential from legitimate users will highly depend on attacker’s experience and skills. If attacker has experience in performing social-engineering attacks, the difficulty of attack step “social engineer internal employee” would be average or low, however if an attacker has little experience in social-engineering, the difficulty of the same attack would be high of very high.

1.1 Motivation

Appropriate conceptual framework that accurately defines security measures, will help to support comparative security assessment and better overview on security threats of the complex system. Majority of quantitative analysis using attack trees, are only focusing on single attribute or parameter, e.g. cost or resources available, assuming their independence of one another. However, in fact interdependent relation between parameters are important. Jürgenson et al. [4] proposal in Parallel and Serial model used sets of parameters e.g. monetary gain if attack succeed, attack costs, attack success probability, probability for of getting caught, and expected penalties. Motivation behind proposing “Attacker Profile” as new factor to be considered in attack tree based quantitative analysis, is that we often have difficulties making estimation of parameters e.g. difficulty, time, success probability because these values are dependent on specific properties and characteristics of the considered malicious actor. This has leads to the idea of the separation of an attack tree as a hierarchical description of attack scenario, and an attacker profile as malicious actor who acts in this environment.

1.2 Scope

This thesis utilizes ApproxTree, by Lenin – Willemsen [17] - a security modeling approach which relies on the attack tree-based quantitative security assessment methodology of Jürgenson et al.
model [4]. In addition, “Attacker Profile” as separate factor in case study attack tree of purchase flow of virtual goods will be highlighted and derived conclusion based on the results.

1.3 Aim
The aim of this thesis is to provide a philosophical point of view on whether the attacker profile is a useful concept as an additional attribute value for the quantitative security assessment, providing a case study which will be used to explain in a more distinct way, if an attacker profile is efficient and give benefit to the final results of analysis.

1.4 Summary
There are only several case studies that being used to measure live system security. Therefore, we will provide insight on attacker profile as additional separate parameter that is implemented in ApproxTree [17]. We compare the result of analysis with set of attacker profiles and without attacker profile and conclude the effects of profiling on attacker’s final outcome.

1.5 Outline
This thesis is organized as follows. Chapter 1 gives introduction to thesis and gives reason, scope and aim of proposing attacker profiling in quantitative security measurement, and utilizes ApproxTree [17]. Chapter 2 we describe the introduction to security modeling in general. Chapter 3 gives an overview of the formal definition of attacker profile, Jürgenson – Willemsen parallel model logic, list of parameters involved and also computational logic for ApproxTree as calculation program [17]. Chapter 4 describes in detail an attack tree “Fraudulent Purchases in Digital Goods” that we used as the case study. Chapter 5 provides the calculation result using ApproxTree [17] and discussion about the result is also presented here. Chapter 6 gives final conclusion based on calculation results outlined in Chapter 5, and also proposes suggestions for future improvements involving attacker profile in quantitative security assessment research area.
Chapter 2
Introduction to Security Modeling

2.1 Concept of Security Modeling

For the past decade, many companies and commercial organizations have claimed information to be one of their valuable assets, have put tremendous effort in implementing appropriate security policies according to ISO-27001 and ISO-27002. Security policies are needed in compliance to standardize ISO for protecting information from unauthorized disclosure, access and modification of information itself. Therefore it is needed to measure and analyze commercial systems that aim to improve scientific and systematic engineering method against various threats or vulnerabilities.

Studies of computer security has been evolving for the past two decades. Various different studies in security modeling expands in complexity, sophistication, and relevance with real life conditions. Computer security experts and scientists try to address problem, flaws, potential threats, leakage that arose in real life conditions, in existing systems or networks. Security assessment of a system involved precise definition of security, question on “how secure is the system?” is heavily dependent on components that build the very system intact, properties, network, and human factors.

Security engineering is a relatively new field in computer world, however there are several papers that discussed general security modeling in early 2000-s, i.e [6] Schumacher and Roedig, and book by Schumacher [5] published in 2003. They highlight the importance several security patterns that can be implemented to facilitate more systematic and effective way in security engineering processes, for example combination between security patterns and attack tree as representation to identify threats, or attacks in more organized manner.

Furthermore, Miede et al. in their paper [7] suggest on the use a meta-model for IT security, demonstrate using concept of attack factors like attack execution phases, attack type, proficiency of attackers, attack action, attack tools, and more importantly underline the use of qualitative and quantitative attack metrics to measure the outcome of the attack itself. This proposal is backed up by several researchers like Trivedi et al, their paper [8] mentioned about combination of dependability and security model also presented case studies using fault trees and attack trees.
Nicol et al. [12] highlights that system security model has more important attributes such as, confidentiality, non-repudiation, attackers behavior, that are difficult to be measured using standard, stochastic techniques.

Graphical security models are one of the systematic methodologies used as threats and risks evaluation of real-life systems by researchers and also adopted by commercial industries. Attack trees as one of many graphical security models are the most usable among graphical attack modeling techniques, first pioneered by Fault Tree were introduced back at 1981 by Vesely et al. in form of Fault Tree Handbook [13], the revised version of which published by NASA in 2002. Weiss [10] introduced threat logic trees as the first graphical attack modeling technique that have obvious similarity with fault trees [10]. Weiss approach is the base for various graphical security model nowadays. There are important points from Weiss approach, first simple procedure for quantifications of attack tree with bottom-up algorithm, different functional operators for combining the value of child nodes in node hierarchy, shown in Figure 1. The leaf nodes could be extended with attributes e.g. cost levels or skill level or behavioral indicators.

![Diagram of a Threat Logic Tree](image)

**Figure 1. Example of Threat Logic Tree, taken from Weiss [10] paper: Unix System Design**

In terms of quantification analysis of security model, Schneier [1] proposed various additional values to attack tree nodes which called attributes, value of attributes are not limited to attack costs, attacker skill level or proficiencies, attack time and impact of attacks. Many researchers have
consider Schneier proposal as basis to expand attack trees and its attributes, such as Amroso [14], Mauw and Oostdijk [2], Buldas et al. [3], Li et al. [15], and Tanu and Arreymbi [16], each of these researchers has shown various ways to populate attribute values and derived final results from complex attack tree.

There has been vast case studies such as [18,19,20,21,22,23,24,25], applying attack tree approaches in various security analysis. Attack tree has been improved since first introduced by Weiss [10]. Many research projects, Master, Ph.D Thesis and also real-life system of commercial industries have important role on contributing attack tree extensions to more sophisticated and complex attack models for example, [25,26,27,28,29,30]. Initial development of attack tree analysis is not limited to quantification of attributes of attack tree models itself, but also attained on commercial software for attack tree modeling. AttackTree++ from Isograph [31] with various attack tree template to be offered, SeaMonster as an academic tool as result of SHIELDS projects [32] which provides sophisticated visualization and library support, SQUARE tool [33] with specific SQUARE methodology [34], SecureITree [35] from Amenaza, and AttackDog [36] which applicable as system security risk evaluation software.

The number of research projects utilizing attack tree analysis in national or international scale, along with Master and PhD theses contributions are still growing. Thus, attack tree analysis certainly will still be prominent graphical method to be developed and explore its potential for wider usage of real-life commercial systems.

2.2 Attack Tree Analysis

The earliest encounter on graphical attack modeling techniques was introduced by Weiss et al. in 1991 [10]. However, Schneier papers [1,11] are the most cited for attack tree development studies. Majority of researchers mentioned that threat logic trees have similarity with the fault tree analysis which was introduced by U.S. Air force, Boeing and nuclear power industry in 1960’s and 1970’s into a comprehensive handbook [13], before it was revised and published by NASA in 2002 [36].

Attack tree analysis demonstrate systematic method to specify threats toward a system, in form of a tree structure. Within the tree structure, the final goal of an attacker is represented as a root node of the tree, will be refine into smaller and specific attack schemes, known as leaf nodes or child
nodes. Leaf nodes are the representation of specific various ways of how the parent nodes are achieved. Attack tree has two different type of nodes, AND – nodes and OR – nodes. AND – nodes are tree elements which are satisfied if all child nodes are satisfied likewise, OR – nodes are satisfied if any of the child nodes is satisfied. Many researchers have implemented, exhibit various values or attributes or metrics on leave nodes, depending on how precise the final result is expected by the researcher. According to Schneier [1], most common attributes used in an attack tree are cost, probability, severity, impact and consequence. Scheneier also underlined that attributes can be standalone value or combined to determine the cheapest attack path.

![Attack Tree Diagram]

**Figure 2. Example Simple Attack Tree assigned with value I = Impossible ; P = Possible, Schneier [1]**

Kienzle et al. in their paper [37] also proposed general procedure for attack tree constructions. Furthermore, [38] by Morais et al. have developed a top-to-bottom process to create attack tree for security protocol analysis as below listed:

- The root level represents the final goal of attacking security protocol target.
- The second level represents security properties that attacker attempts to violate.
- The third level represents the method being exploited by the attacker, as for example, key exchange, message truncation, etc.
- Subsequent levels represent attack stages to achieve third level sub-goals.
From vast and diverse studies on attack tree method with basic AND – nodes and OR – nodes, are still potential to be studied and evolved towards new other models, like defense, threat, vulnerabilities.

2.2.1 Attack Graph

In 1998, Philips et al. [39] introduced the attack graph concept as representation of probable attack states of a system. They highlights 3 important inputs for an attack graph:

- Pattern of Attack, including all general information of attacks stored in a database.
- Detailed system properties information, including network topologies used in the system, system configurations, etc.
- Attacker Profile, including skill or proficiency, tools and other resources.

Thus result of attack graph analysis for a system could form set of an attack path, with probability shortest low-cost attack path. Papers which use the attack graph concept, published in 2001 – 2005 were gathered into [40] mostly focuses on enriching attack graph with various parameters, calculation methods, and specific analysis in association of several additional aspects.

Furthermore, set of papers on attack graph related to quantitative assessment of networked system are also available [41,42], new proposed metric in an attack graph, such as attack resistance [43], and probabilities [42]. Moreover, Wang et al. [42] refine attack graph with probabilistic metric intended for efficient computational purpose. Noel et al. [44] exhibits simulation of network penetration and performs assessment of an overall security of the system by propagating attack likelihoods, this simulation is used to identify possibilities of combinations of multiple vulnerabilities to form an attack towards a networked system. Result of simulation can be derived as a score level of risk mitigation which can be used for effective strategy lowering the cost and having high impact of system security.

Dantu et al. in [45] added attacker behavior to the attack graph metrics, emphasis on calculation using Bayesian methodology to calculate risk level of critical resources. Attack graph itself has been developed for different goals to assess and measure various aspects of network security [42]. Similar proposal [46], utilize deductive tree structure which used inductive reasoning to excerpt final expected goal.
2.2.2 Attack Tree Application

Attack trees have been used in numerous studies and research projects. Some of them are focused on utilizing attack tree analysis as representation of threat modeling in real-life systems. Fung et al. in [23], mentioned about a usage of attack tree for survivability study, they form intrusion scenarios and compute attacker cost function to execute the scenarios. Fung et al. principal [23] is the system survivability was defined by the weakest link in all the intrusion scenarios, although the quantitative analysis model itself as consistent with Mauw et al. foundations [2], but Fung et al. model [23] only considers single parameter ‘attack difficulty’ which is not adequate, according to Becker [9].

Attack trees were also used to analyze real-life systems, such as BGP routing protocol [50], Security of mobile TAN on smartphones [51], online banking system [54], RFID networks [55], SCADA system [52,53,16], nuclear control system [56], security analysis digital content copyrights protection system [57]. This is one of many attack tree applications that are used for assessing the security of distributed system. In addition Mägi thesis [58] in collaboration with Buldas, utilizes attack tree to assess between two e-voting schemes, Estonian E-Voting system and USA Secure Electronic Registration and Voting Experiment (SERVE) in 2004.

2.3 Attack Tree Attributes

Quantitative analysis of security model, are relies on various attributes. There are common attributes in attack tree quantitative analysis, for example: costs of attack, while other attributes are quite scarcely used in attack tree.

2.3.1 Concept of Attack Tree Attributes

First concept of populating leaf nodes of attack tree with specific values, such as costs, probability, severity, impact and consequence which represents component’s cost to conclude low-cost attack path was introduced by Schneier in 1999 [1]. All of these attributes can be individually calculated or combined, later followed by so many researches and propose improvement on the basic model of attack tree attributes.

Various researchers have come up with new types of attributes, Buldas et al. [3] propose attribute called expected attacker’s outcome or known as attacker’s utility, this attribute describes monetary
global gain of attacker, derived from several parameters like gains of the attacker if attack succeeds, costs of attack, probability, probability of getting caught and penalties in case of being caught. Henniger et al. [49] introduced attack time parameter, similarly labelled by Çamtepe et al. [47] as - time to live, this means extent of time which is needed to perform attack and might be dependent on attacker’s proficiency and resources. Byres et al. [48] propose detectability that describe how sophisticated attacker skills and techniques. Attributes for example like electricity, or insider help can be added as conjunctive value.

Kordy et al. [60], highlight the importance of meta-attributes. The values that are assigned to attributes need to be elaborated to distinct components for accuracy, although it might have the same value. For example, ‘confidence’ represent degree of certainty of decorator toward specific value on that child node. This value can use percentage or range descriptor. In addition, ‘coverage’ represent set of number or area that can be associated or effected by this specific attribute value in child nodes.

Kordy et al. [60] also mentioned about value domains. The most common value of attribute is Boolean. Attribute value domains quantified with Boolean values are more suitable for hypothetical schema. Values can be determined by huge range over diversified types of mathematical domains. For example, ordinal scale like Low, Medium, and High, realistic numbers or discrete or continuous probability distributions. Value domain can be applied onto meta-attributes also. For examples, attacker probability of success may use percentage 0 % - 100%, attack costs can either use real numbers or value unit like {low;medium}, as for meta-attributes confidence or coverage could use nominal scale from 1 to 5.

Furthermore, values that were assigned to non-refined nodes or leaves are sometimes determined with uncertainty. Estimating parameters of leaves of attack tree is often difficult, because attack steps performed by an attacker are affected by attacker’s skills and resources. Thus we need to consider profiling for malicious actors resources and skills, which affect sub-goal outcome in attack tree.

2.3.2 Quantitative Analysis Calculations

This section will provide calculation method that previously introduced and developed by researchers in the literatures. Attributes are the main ingredient for quantitative analysis of attack
trees. The value of attributes should be easily understandable for quantitative calculation, thus applicable to complex attack tree with numerous attributes and easily automated.

Mauw et al. [2] was first to present formal method for attributes in attack tree with semantics approach. However Kordy et al. [61] proposed to combine several attributes value to derive the value of attributes which are difficult to be estimated. For example, Jürgenson et al. [4] consider following parameters: gain of the attacker if attack succeeds, costs of attack, success attack probability, probability of getting caught and penalties in case when an attacker is being caught, as attributes value expected attacker’s outcome. Edge et al. [62] present formula for risk attribute on attack tree as below:

\[
\text{Risk} = \left( \frac{\text{Probability}}{\text{Costs}} \right) \times \text{Impact}
\]

Henniger et al. [49] also uses the combination of elapsed time, expertise, knowledge of the system, window of opportunity and required equipment, in order to deduce the required attack potential.

### 2.4 Boolean Formula

True quantified Boolean formula (TQBF) based on source [63] is a formula in quantified propositional logic which the used quantified variable, either existential or universal quantifiers. This formula is equivalent to either true or false condition. For example we have simple Boolean function example as \( F = X_1 \lor X_2 \), the result of Boolean formula will return True, with at least one of the quantifiers is true and will return false when both of the identifiers are false. If we have created attack tree, then it is certainly corresponds to a Boolean function.

In case if our attacker has planned available with set of elementary attacks \( X \), then he needs to figure it out which subsets of \( X \) are sufficient or profitable to be executed, to achieve the root node. In this thesis, we assumes that the set of subsets is monotone, if some random set of elementary attacks is suffices, so does any of its supersets. This approach is chosen because it is a convenient way to describe all the successful attacks by a monotone Boolean function \( F \) on the set of variables \( X \).
Chapter 3  
Related Work  
Implementation of attack tree in various research projects, Masters and Ph.D Thesis, has been mentioned in the previous chapter, however this chapter proposes additional parameter in attack tree, called “Attacker Profile”. Attacker properties used in attack tree for quantitative analysis have been used in several researches, there are several papers that actually highlight the importance to have insight on attacker skills, resources, behavior, motives, detectability, attack complexity, quantity of attacker, and many more, thus can be used to deduce the final value of refined nodes.

Attacker profile concept in security assessment has been research topics in past studies. However, specifically attacker profile in research area of quantitative security assessment is very few. In this thesis we propose attacker profiling as independent approach towards attack tree as quantitative security assessment. Aim for this section is to give a clear description about what attacker profile is. Derived from parameter combination, the value estimation, and also the ApproxTree calculation program [17] to generate the result on chapter 4.

3.1 Attacker Profile Overview  
Back in 1998, Philips et al. [39] already proposed attacker feature in attack graph as one of important point as attack graph input. Several research projects have focusses on attacker profiling using honeypot in “Know Your Enemies” series [75 - 77], have identified attacker techniques, tools, and targeted systems. Furthermore, several researchers proposed concept of attacker personas, which is related to goal, motivations, attitudes, and skills [70 - 73], Shamal et al. underlined that in papers [72], that attacker personas are only based on assumptions. They also highlight insider threat motivations and characteristics, as well as the use of attacker persona to identify and exploit threat. Collective or individual attacker has been main interest of some researchers [64 - 69] and heavily related to attackers skill, motivation, ideology, goal and interest.

As for quantitative analysis which involved attacker profiling in their paper, Harold et al. [74] has used threat tree and Monte-Carlo simulation to perform risk assessment of a voting system. They mentioned the importance of attacker characteristic and also the complexity of the attack itself, previously Jones [78] analyzed voting system in 2005, using the same simulation made bold point
that likelihood of an attacks can be referred as “cost” of an attacker, which can be estimated using various different ways, such as dollars, number of people, and time effort. Jones [78] and Tipton et al. [79] has also included risk aversion, degree of difficulty, discoverability, ease of access, effectiveness of controls, effort, incentive, interest, skill level, motivation, resources required, risk of detection, and special equipment needed as factors that can be included in attacker profiling.

From various literature outlined in previous paragraphs, there are some common parameters that are more often used in research projects to define an attacker profile, these values are more feasible for quantitative analysis and gives clear understanding of attributes corresponds to attacker profiling. We propose attribute “Attacker Profile” as combination of several parameters e.g. skills or proficiencies, budget, and time required for performed attack steps. For more detail on these parameters which included in “Attacker Profile” on ApproxTree [17] will be presented in the next section.

3.2 Jürgenson - Willemsen Parallel Model Logic

The description of parallel model presented by Jürgenson – Willemsen, its logic will be explained in this section. Jürgenson – Willemsen model [4] computational outcome is relies on individual attack suites, below are the following logic of Jürgenson – Willemsen model [4].

1. Elementary attack is the bottom level abstraction of attack tree, it does not have internal structure and should be referred as possible available to a specific attacker actions. Elementary attacks also known as leaves of attack tree.

2. Attack tree $T$ is a simplification of a PDAG structure $(V = N \cup X, n_0, E)$, of the following elements:
   a. The set of leaves $X = \{X_1, \ldots, X_n\}$ corresponds to elementary attacks with specific parameters assigned to them.
   b. The set of nodes $N = \{N_1, \ldots, N_m\}$ correspond the logical function of AND ($\&$) and OR ($\vee$). The AND function will be evaluated to true if all the child nodes evaluate to true. OR function however, will be evaluated to true if at least one of the child nodes evaluates to true.
   c. $n_0 \in N$ is the root node as attacker final goal, it does not have any parents.
3. Attack suite \( S \subseteq X \) is set of elementary attacks, or also called attack steps or attack path which is chosen by an attacker to be performed in order to achieve the attacker goal.

4. We say that the attack tree \( T \) is satisfied by an attack suite \( S \) and the goal of attacker is achieved if the Boolean function represents root node \( n_0 \) evaluates to true when all elementary attacks from the attack suite \( S \) have been tried and they have evaluated to true and false values.

The outcome value for attack suite \( S \) can be computed with formula shown below:

\[
\text{Outcome}_S = P_s \cdot \text{Gains} - \sum_{X \in S} \text{Expenses}_i
\]

Genetic algorithm is presented with following set of rules. First, generate initial populations of 20 individuals (randomly), checks if they are satisfying \( F \) and the value of \( F \) is True, second crossover individuals with themselves and create new population (new generation). Third, mutate some individuals with probability value 10\% (value 0.1) and last step is join the mutated individuals with the original individuals, then after that proceed with computing the outcome and choose the best attack steps.

3.3 Attacker Profile Parameters

As suggested by Lenin – Willemsen [17], attacker profile consists of value from several parameters which describe attacker/s skills and resources available for performing malicious task and goals. Attacker profile parameters choice was based on common parameter value that were used in majority of researches to describe attacker behavior, resources, skills of an attacker in a quantitative analysis. “Attacker Profile” is related to several parameters: skills or proficiencies, time resource, and attacker’s budget. Below are descriptions on parameters:

1. **Attacker Skill or Proficiency**, this parameter is related to attacker’s skill level on performing an attack, this include techniques that attacker choose to perform the attacks.
   Value unit used for attacker skills: Very High (V), High (H), Medium (M), and Low (L).

2. **Attacker Budget**, this parameter is related to cost resources (monetary) which includes hardware or software resources that are used to support or perform the attack steps, thus need to be converted in the monetary unit. Value unit used for Attacker budget: in specific currency value.
3. **Time available for attacking**, this parameter is related to the range of time available for attacker to perform specific attack step. Value unit used for Time availability: Second (S), Minutes (MT), Hours (H), Days (D).

### 3.4 Attacker Profile Formal Definition

Attacker properties have been part of consideration of various research projects. Attacker profile components use the following parameters: skills or proficiencies, motivation, cost required, complexity of attack, individual or collective attackers, time resource, and attacker’s relation toward targeted system (insider or outsider attacker). According to Lenin – Willemsen [17], attacker profile formal definition as following:

Definition 3.4.1: An attacker profile is a Boolean function, considering a single attack suite as input parameter and returning true, if the attack suite matches profile constraints as stated below, returning false otherwise.

1. The sum of costs of each elementary attack in S, is smaller or equal to than attacker budget
   \[ \sum_{i} \text{cost}(X_i) \leq \text{Budget} \]
2. For all elementary attacks in an attack suite the level of difficulty has to be easier or not greater than attacker skills
   \[ \forall (X_i) \in S \leq \{ \text{Difficulties} (X_i) \leq \text{Skill Level} \} \]
3. The attack execution time of each attack step is lesser than the attacker time allocation
   \[ \sum_{i} \text{time}(X_i) < \text{Attacker Time Allocation} \]

Based on definition 3.5 we provides example, there are attacker profile 1 (X_i) with budget = 100, skill = Medium, time allocation = Days and attacker profile 2 (X_i) with budget = 0, skill = High, time allocation = Hours. Attacker profile 1 will suite to perform the “trick owner” attack because attacker do not have high skill to perform the “threat owner” attack, likewise attacker profile 2 will suite to perform the “threat owner” attack because attacker does not have appropriate budget nor time allocation.
3.5 Attacker Profile and Parallel Model of Multi-Parameter Attack Trees

Parallel model for Multi-parameter attack tree considers rationality assumption, which means that attacker will start to launch an attack if the attack scenario is profitable, otherwise the system is considered secure against rational profit-oriented attackers who want to achieve maximal outcome. Parallel model approach uses the same parameters with Buldas et al. [2] model, and improves it with assumptions that all elementary attacks will be performed simultaneously, thus the attacker does not consider his/her decisions on success or failure of some elementary attacks or leaf nodes of attack tree. In addition, Jürgenson et al. parallel model [4] assumes that the attacker makes decision for each leaf nodes with below rules:

1. **Cost**, attackers has to spend financial resources that he/she owns to prepare and perform attack steps. We use value in numeric as for monetary.
2. **Probability of success** from attack step that will be taken, value used is in range $[0, 1]$.
3. **Level of difficulty**, attack step will be categorized in range value Low/Medium/High/Very high, for attacker to be able succeed obtain sub-goal.
4. **Time required to execute an attack step**, this parameter is highly dependent on the level of difficulty. Value used for this parameter are Seconds/Minutes/Hours/Days.

Figure 18. Attacker profile invalidate nodes and form an appropriate Attack Suites
Attack tree \((T)\) is satisfied by the attack suite \(S\) and the attacker final goal is achieved if the Boolean function corresponds to the root node \(n_0\) evaluates to true and all elementary attacks from attack suite \(S\) have been executed and evaluated to true and false values, in accordance if the elementary attack was successful or failed. We considers monotone Boolean formula, the notion of Boolean function \(Ƒ(S := \text{true}) = \text{true}\), means that elementary attacks \(X_1 := \text{true}, \ldots, X_n := \text{true}\) always evaluates \(Ƒ\) to true.

The effect of applying attacker profile to parallel model Jürgenson et al. model [4], allows us to invalidate certain child-nodes or leaves of the attack tree which leads to elimination of certain sub-trees of the overall attack tree. Therefore, attack tree result which describes the attack scenario affordable to a particular attacker described by an attacker profile. For instance, in Figure 18, attacker goal is stealing a car, for attacker with AP (attacker profile) 1 with budget = 100, skill = Medium, time allocation = Days, appropriate rational assumption is execute “trick owner” attack suites and invalidate “threat owner” because AP1 doesn’t have suitable skill to perform “threat owner” attack. AP (attacker profile) 2 with budget = 0, skill = High, time allocation = Hours, is suitable to performed “threat owner” invalidates “trick owner”, because attacker profile doesn’t pass the validation on budget and time allocation.

### 3.5.1 ApproxTree Calculation Method

Jürgenson – Willemsen model [4] logic outlined in section 3.2, has been used as basis of AproxTree calculation method, proposed by Lenin and Willemsen [17] in late 2013, has all the algorithm and calculation method of Jürgenson – Willemsen model [4]. As for attacker profile validation towards attack suites, such as comparison attack suites cost against attacker budget, attack suites level of difficulty against attacker skills or proficiency and time to execute attack suites against attacker time allocation (refer to Definition 3.5), then followed by compute the outcome using same formula below, and chooses the maximum outcome from several result of calculations:

\[
\text{Outcome}_{Si} = P_{si} \cdot \text{Gains} - \sum_{X_i \in S_i} \text{Expenses}_i
\]

\[
\text{Outcome} = \max \{ \text{Outcome}_{Si} : S_i \subseteq X, Ƒ(S_i := \text{true}) = \text{true} \}.
\]
Chapter 4

Case Study: Fraudulent Purchases of Digital Goods

Over more than two decades, internet has transformed ways people doing business for instance, on-line transactions on digital goods, online banking system or e-wallet services, social networks, and many more. Security issues has become interesting topics in designing online payment systems. There are always standard network security schemes that can be implemented towards the design, like data encryption and firewalls. However, this approach is not sufficient to prevent dynamic nature of fraud threat towards online payment systems. Therefore, financial institutions and online merchants have developed systems that can detect fraud on their online payment component like credit card and e-wallet using approaches like neural networks, data mining, meta-learning, game theory, and support vector machine.

Fraudsters are likely to have an adaptive behavior and within time can compromise any protection mechanisms. As preventive systems are developed, so does the criminal technologies will evolve toward counter protection mechanism to speed up their fraudulent activities. Based on these issues which financial institutions and online merchant encounter, we choose to present this case on fraudulent purchases on digital goods, for example; Skype credit balance, Amazon digital books, games token, vouchers, and many more. This case study is emphasizing fraudster profile on launching their attacks to obtain legitimate customers payment detail and proceed with fraudulent purchases of digital goods. Highlights on this case study are based implementing fraudster profile towards attack tree and utilizing ApproxTree [17] calculation program.

4.1 Attack Tree Model

The background given in previous section served as a preliminary description to create an Attack Tree that we then used for attribute decorations. Attacker motivation is to solely obtain maximum profit from abusing or making fraudulent efforts with less or even zero budget. Payment flow in well-known websites that sold digital goods or services can be as perfect as it seems, but as most of architecture that involving several third party redirection sites, the more complex payment redirection method will be prone to malicious users spotted flaws by learning the payment flow pattern.
The final goal of the attacker is fraudulent purchases of digital goods, shown in figure 3. In order to achieve final goal, the attacker has 3 options. He or she can choose either “victim account take over”, meaning that the attacker goal here is to compromise victim username and password of that particular website, or “usage policy violation”, meaning that the attacker intended to break the limitation and exploit the opportunities lying behind the usage policy of a particular product, or “compromised CS (customer service) agents account”, meaning that the attacker targeted an attack towards specific individuals who are working inside the organization and compromising customer service agents credentials. Boolean function of attack tree “Fraudulent Purchases of Digital Goods” as below:

\[ F = X_{39} + (X_1 + X_2) + (((X_3 + X_4) + (X_5 + X_6)) + (X_7 + X_8) + X_9) + (((X_{10} + X_{11}) + X_{12}) + (X_{13} + X_{14} + X_{15})) + (X_{16} + (((X_{17} + X_{18}) + (X_{19} + X_{20} + X_{21})) * X_{22}) + (X_{23} + X_{24})) + (((X_{25} + X_{26}) + (X_{27} + X_{28} + X_{29})) + (X_{30} + X_{31})) + (((X_{32} + X_{33}) + (X_{34} + X_{35} + X_{36}) + X_{37} + X_{38})) \]

The complete refinement are represented by figure 4, figure 5 and figure 6. Figure 4 represents refinement if attacker chooses to “take over account from legitimate victim”. There are several ways to be choose, for example: “Blackmail victim”, “Social engineer victim to reveal account data” which can be done by “quid pro quo” for instance, making a fake phone calls that impersonating to be a technician and asking more detail about account and password to victim, or asking victim to type some command that can enable attacker to gain privilege escalation, in addition to “quid pro quo”, attacker can also launch phishing e-mail toward targeted victim.
If attacker chooses to “Infect victim PC with Malware”, refined child nodes are shown in Figure 7. There are several options that can be launched by an attacker, by “physical” access directly towards victim’s PC or Laptop, or “pose as PC technician imposter” in order to have direct contact toward victim’s PC which enable attacker to insert malicious USB contain backdoor or stealth key-logger. Other options for attacker is perform “social engineering”, like “spear-phishing” or also “lure victims to malicious website” which will infect the victim with malicious software automatically. In addition to options that are available, another option which can be used by attacker is “web obfuscation”, the attacker can plant malicious script into a legitimate website and finally automatically download unwanted malicious software to victim’s PC without victim consent.

The other method that attacker can use is “shoulder surfing”, this method has to be done in combination with physically spying on the victim, along with learning ID also password details and getting payment method to finance the fraudulent purchases through illegal means like “threat victim to reveal CC data”, “steal CC data” or “buying illegal stolen CC data from underground hacker’s forum” as shown in Figure 8. In addition to options on “victim account take over”, is “sniffing network traffic” method. If attacker chooses to launch this method, the attacker is considered having more technical knowledge or having enough budget to employ a person who capable of tapping and hacking techniques on wireless router to gain ID and password. Moreover “wired tapping” will need more time, as “apply job to ISP company” will need to wait until attacker got the job. “Remote tapping” will requires attacker to have a very high until high skill on hacking techniques and considerably sufficient time to penetrate the ISP system defense, we assumes all ISP companies are well-equipped with security countermeasure towards hacker attacks.
The following refinement of child nodes shown in Figure 5, is related to internal policy of the website store, every website selling the digital goods, has its own policy that state clearly in details the rules of games, product details and restrictions, policy on the usage, and digital rights infringement. These usage policies are usually available in separate section, or FAQ page, or when user is signing up for membership of particular website. “Create new account” is almost effortless, most of the websites treat the user input equally. However, “Report account blocked poses as
legitimate user” also doesn’t require much technical knowledge, if attacker happens to know a victim (for example email, username, or software type) he/she can lie to customer service and ask to unblock the specific account for attacker to gain full control over legitimate victim account, also it needs to be combined with options for getting the payment method to finance the fraudulent purchases. Moreover, refinement of child nodes “bypass usage policy” are related to the attacker effort to propagate the profit with less budget to get more benefit from products, for example attacker can make resells the product from unofficial or unauthorized websites which offers the same exact product, or for example resells printed books (if attacker was buying digital books) to get profit out of the original copy.

Figure 9. Refinement “Sniffing Network Traffic”
Human factor in a system might be source of flaws, every customer service agents had undergone several trainings and follow company policies to deliver their service to customers. However this cannot guarantee that customer service agents are tamper-proof towards directed attack. As most of customer service agents are having access towards the internal system, gaining their access could result in a greater benefit to the attacker. In addition, if an attacker can elevate customer service agents privilege to higher access, this can lead to even more destructive effect on internal system. Refinements shows in Figure 6 has several options for attacker to choose with aim compromising customer service agents account to make fraudulent purchases.
4.1.1 Attacker Profile: Types of Actors Involved

We create attack tree with assumption only one player as attacker is involved, below we provide different attacker profiles which allow us to perform analysis of various attacker types:

a. **Attacker Skill**, value unit used for attacker skills: Very High (V), High (H), Medium (M), Low (L).

b. **Attacker Budget**, value unit used for Attacker budget: Numeric value in EUR.

c. **Time available for attacking**, value unit used for Time availability: Second (S), Minutes (MT), Hours (H), Days (D).

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker Profile 1</td>
<td>2000</td>
<td>V</td>
<td>HR</td>
</tr>
<tr>
<td>Attacker Profile 2</td>
<td>500</td>
<td>V</td>
<td>D</td>
</tr>
<tr>
<td>Attacker Profile 3</td>
<td>1000</td>
<td>H</td>
<td>MT</td>
</tr>
<tr>
<td>Attacker Profile 4</td>
<td>200</td>
<td>H</td>
<td>MT</td>
</tr>
<tr>
<td>Attacker Profile 5</td>
<td>50</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Attacker Profile 6</td>
<td>750</td>
<td>M</td>
<td>D</td>
</tr>
<tr>
<td>Attacker Profile 7</td>
<td>50</td>
<td>L</td>
<td>D</td>
</tr>
<tr>
<td>Attacker Profile 8</td>
<td>1500</td>
<td>L</td>
<td>HR</td>
</tr>
<tr>
<td>Attacker Profile 9</td>
<td>0</td>
<td>V</td>
<td>S</td>
</tr>
<tr>
<td>Attacker Profile 10</td>
<td>0</td>
<td>V</td>
<td>MT</td>
</tr>
<tr>
<td>Attacker Profile 11</td>
<td>0</td>
<td>V</td>
<td>HR</td>
</tr>
<tr>
<td>Attacker Profile 12</td>
<td>0</td>
<td>M</td>
<td>S</td>
</tr>
</tbody>
</table>
We present common possible combinations for each of the parameters available, which relate to real-life conditions. Although, these total 19 profile are not representing all spectrum of attackers available, however we need to highlight that final gain or profit that is set to 10000 EUR, therefore we assume that it is unlikely an attacker will invest into an attack more than the quarter of the final gain or profit.

### 4.2 Attack Tree Decoration with Attributes

In this particular case study, the chosen attributes correspond to the same multi-parameter attributes that were used by Buldas et al. [3] and Jürgenson et al. [4] selected attributes for our initial case with detailed descriptions, and corresponding value domains can be found in Table 2 below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker Profile 13</td>
<td>0</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Attacker Profile 14</td>
<td>0</td>
<td>M</td>
<td>HR</td>
</tr>
<tr>
<td>Attacker Profile 15</td>
<td>0</td>
<td>L</td>
<td>S</td>
</tr>
<tr>
<td>Attacker Profile 16</td>
<td>0</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Attacker Profile 17</td>
<td>0</td>
<td>L</td>
<td>HR</td>
</tr>
<tr>
<td>Attacker Profile 18</td>
<td>0</td>
<td>L</td>
<td>D</td>
</tr>
<tr>
<td>Attacker Profile 19</td>
<td>10000</td>
<td>V</td>
<td>D</td>
</tr>
</tbody>
</table>
Table 2. Decoration attributes for “Fraudulent Purchases of Digital Goods” Attack Tree

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costs</td>
<td>Budget that relates to expenses that were used by attacker to finance the attack. For instance, hacking development tools, specific hacking software, illegal information, hiring criminals for specific job, fake properties and sum of bribe.</td>
<td>Estimated Numerical value</td>
</tr>
<tr>
<td>Gain</td>
<td>If attacker has successfully reached the final goal, thus will result in economic gain or profit that attacker receive.</td>
<td>Estimated Numerical value</td>
</tr>
<tr>
<td>Probability</td>
<td>Presumably chance that the attack will succeed. Could be based on heuristics of similar attacks or cognitive estimations.</td>
<td>Specific numeric value Range [0……..1]</td>
</tr>
<tr>
<td>Difficulty</td>
<td>The level of technical expertise or proficiency along with social skills of attacker that will be needed to carry out the attack in order to succeed.</td>
<td>Very High (V) : Beyond the known capability of todays best attackers High (H) : Requires high degree of technical expertise and lots of experience, usually criminal cracker/hacker by profession. Medium (M) : Requires a bit technical knowledge, lack of experience, and heavily depend on available hacking tool resource. Low (L) : Doesn't need to have technical skills to perform the specific attack.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Values</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>Duration of time that needed for attacker to perform the attack, apart from difficulty and costs of attack.</td>
<td>Days (D)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hours (H)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minutes (MT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Seconds (S)</td>
</tr>
</tbody>
</table>
Chapter 5
Observation and Discussion

Proceeding the use case study of “Fraudulent Purchases of Digital Goods” on chapter 4, we continue to apply estimated values for each parameter of the child nodes in Table 3. The estimation values for each attributes were chosen based on history of similar attacks or cognitive estimations, and by adopting ApproxTree [17] as based of our calculation method to be combined with an attacker profile and estimation of quantitative parameters for leaf nodes of attack tree needs to be estimated in an actor-independent way as much as possible while constructing the refinement of attack tree on our case study “Fraudulent Purchases of Digital Goods”. Furthermore we proceed with performing calculation using ApproxTree [17], included input files with details estimation parameters value in Table 3.

Table 3. Estimation Parameter value for child nodes “Fraudulent Purchases of Digital Goods” Attack Tree

<table>
<thead>
<tr>
<th>Child Nodes Name</th>
<th>Attack Suites Name</th>
<th>Budget</th>
<th>Probability</th>
<th>Difficulty</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quid pro quo</td>
<td>X1</td>
<td>50</td>
<td>0.30</td>
<td>M</td>
<td>HR</td>
</tr>
<tr>
<td>Phishing Victim</td>
<td>X2</td>
<td>50</td>
<td>0.45</td>
<td>M</td>
<td>HR</td>
</tr>
<tr>
<td>Break into premises</td>
<td>X3</td>
<td>150</td>
<td>0.5</td>
<td>H</td>
<td>HR</td>
</tr>
<tr>
<td>Stole PC</td>
<td>X4</td>
<td>50</td>
<td>0.65</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Impersonate PC technician</td>
<td>X5</td>
<td>50</td>
<td>0.65</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Insert malicious USB</td>
<td>X6</td>
<td>0</td>
<td>0.9</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Spear-phishing victim</td>
<td>X7</td>
<td>0</td>
<td>0.75</td>
<td>M</td>
<td>D</td>
</tr>
<tr>
<td>Bait victim to malicious website</td>
<td>X8</td>
<td>0</td>
<td>0.15</td>
<td>M</td>
<td>D</td>
</tr>
<tr>
<td>Web obfuscation</td>
<td>X9</td>
<td>200</td>
<td>0.3</td>
<td>H</td>
<td>HR</td>
</tr>
<tr>
<td>Learn ID</td>
<td>X10</td>
<td>0</td>
<td>0.35</td>
<td>L</td>
<td>S</td>
</tr>
<tr>
<td>Brute-force password</td>
<td>X11</td>
<td>0</td>
<td>0.65</td>
<td>H</td>
<td>HR</td>
</tr>
<tr>
<td>Dictionary attack</td>
<td>X12</td>
<td>150</td>
<td>0.70</td>
<td>M</td>
<td>HR</td>
</tr>
<tr>
<td>Threat victim to reveal CC data</td>
<td>X13</td>
<td>150</td>
<td>0.70</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Stole CC</td>
<td>X14</td>
<td>50</td>
<td>0.65</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Buy CC from underground hacker's forum</td>
<td>X15</td>
<td>75</td>
<td>0.75</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Wireless tapping</td>
<td>X16</td>
<td>0</td>
<td>0.85</td>
<td>M</td>
<td>D</td>
</tr>
<tr>
<td>Child Nodes Name</td>
<td>Attack Suites Name</td>
<td>Budget</td>
<td>Probability</td>
<td>Difficulty</td>
<td>Time</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------</td>
<td>--------</td>
<td>-------------</td>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>Apply for job in ISP company</td>
<td>X17</td>
<td>0</td>
<td>0.50</td>
<td>L</td>
<td>D</td>
</tr>
<tr>
<td>Bribe ISP employee</td>
<td>X18</td>
<td>350</td>
<td>0.40</td>
<td>H</td>
<td>HR</td>
</tr>
<tr>
<td>Tailgating</td>
<td>X19</td>
<td>100</td>
<td>0.40</td>
<td>H</td>
<td>HR</td>
</tr>
<tr>
<td>Phishing ISP employee</td>
<td>X20</td>
<td>100</td>
<td>0.25</td>
<td>H</td>
<td>HR</td>
</tr>
<tr>
<td>Baiting ISP employee</td>
<td>X21</td>
<td>0</td>
<td>0.10</td>
<td>H</td>
<td>D</td>
</tr>
<tr>
<td>ISP traffic tapping</td>
<td>X22</td>
<td>0</td>
<td>0.80</td>
<td>M</td>
<td>HR</td>
</tr>
<tr>
<td>Hack ISP gateway</td>
<td>X23</td>
<td>1500</td>
<td>0.40</td>
<td>H</td>
<td>HR</td>
</tr>
<tr>
<td>Install sniffer</td>
<td>X24</td>
<td>0</td>
<td>0.80</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Create new account</td>
<td>X25</td>
<td>0</td>
<td>0.99</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Report account blocking poses as legitimate user</td>
<td>X26</td>
<td>0</td>
<td>0.75</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Threat victim</td>
<td>X27</td>
<td>0</td>
<td>0.35</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Stole CC data</td>
<td>X28</td>
<td>50</td>
<td>0.65</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Buy Payment data from underground hacker's forum</td>
<td>X29</td>
<td>150</td>
<td>0.75</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Abuse reselling policy</td>
<td>X30</td>
<td>0</td>
<td>0.80</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Abuse product</td>
<td>X31</td>
<td>0</td>
<td>0.75</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Identify corrupt CS Agent</td>
<td>X32</td>
<td>0</td>
<td>0.60</td>
<td>L</td>
<td>HR</td>
</tr>
<tr>
<td>Bribe target</td>
<td>X33</td>
<td>250</td>
<td>0.70</td>
<td>L</td>
<td>MT</td>
</tr>
<tr>
<td>Phishing CS agents</td>
<td>X34</td>
<td>50</td>
<td>0.45</td>
<td>M</td>
<td>HR</td>
</tr>
<tr>
<td>Impersonate CS supervisor</td>
<td>X35</td>
<td>0</td>
<td>0.35</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Baiting CS agents</td>
<td>X36</td>
<td>0</td>
<td>0.75</td>
<td>M</td>
<td>D</td>
</tr>
<tr>
<td>Threat CS agents</td>
<td>X37</td>
<td>150</td>
<td>0.70</td>
<td>M</td>
<td>MT</td>
</tr>
<tr>
<td>Blackmail CS agents</td>
<td>X38</td>
<td>100</td>
<td>0.65</td>
<td>H</td>
<td>D</td>
</tr>
<tr>
<td>Blackmail victim</td>
<td>X39</td>
<td>100</td>
<td>0.65</td>
<td>H</td>
<td>D</td>
</tr>
</tbody>
</table>

With file input based in Table 3, Boolean functions stated above and attacker profile that we are consider which was mentioned in Table 1, we will need to execute ApproxTree program [17] for more than 3 times and choose the maximum value out of the final gain. All the result is presents in appendix Table 4.

### 5.1 ApproxTree Calculation Result

Calculation result from file input Table 3, Boolean function and attacker profile listed in Table 1 is presented in Table 4 (appendix). We can start making observations based on the result given in Table 4. First, there is similarity in terms of the same result for final gain and attack path that were
produced for different set of attacker profiles that we’ve observed. Below in Table 5, for attacker profile 1, 3, 4, 5 with final result 9875.43 EUR, the following attack path for this result are shown in Figure 10.

Table 5. Calculation result for Attacker Profile 1,3,4,5

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker 1</td>
<td>2000</td>
<td>V</td>
<td>HR</td>
<td>Gain : 9875.4345703125 Max AS : {X25,X26,X27,X28,X30,X31,X35}</td>
</tr>
<tr>
<td>Attacker 3</td>
<td>1000</td>
<td>H</td>
<td>MT</td>
<td>Gain : 9875.4345703125 Max AS : {X25,X26,X27,X28,X30,X31,X35}</td>
</tr>
<tr>
<td>Attacker 4</td>
<td>200</td>
<td>H</td>
<td>MT</td>
<td>Gain : 9875.4345703125 Max AS : {X25,X26,X27,X28,X30,X31,X35}</td>
</tr>
<tr>
<td>Attacker 5</td>
<td>50</td>
<td>M</td>
<td>MT</td>
<td>Gain : 9875.4345703125 Max AS : {X25,X26,X27,X28,X30,X31,X35}</td>
</tr>
</tbody>
</table>

Figure 10. Attack Path based on calculation result for Attacker profile 1,3,4,5

Attack path that in result gives final gain 9875.43 EUR can be achieved from options of 2 child nodes refinements. Current attack path are options between compositions on zero budget attack
suites and attack suites that requires some amount of budget to finance the attack suite, range of difficulties of the attack suites are medium until low and minimum time concern is in minutes.

Second result is shown in Table 6 and Figure 11, corresponds to the attacker profile 2. Which gives the final gain 9996.058 EUR. Highlight of this attacker profile is middle range budget, but having deliberately time (days) to execute the attack and also attacker has very high skill or proficiency.

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
</table>
| Attacker 2   | 500    | V     | D    | Gain : 9996.0576171875
Max AS : {X7,X8,X16,X17,X21,X22,X25,X26, X27,X30,X31,X35,X36} |

The attack path in Figure 11, has shown attack suites that contain various elementary attacks that can be chosen by an attacker, final outcome from the attacker profile 2 is higher if compared to the attacker profiles 1,3,4,5. Although the budget of the attacker profile is not high, however the time attacker had is different from previous time that was available to attacker profiles 1,3,4,5. Variant of attack suites which were available are correspond to level of skill that attacker profile 2 had.

However, calculation result of the attacker profile 6 shown in Table 7 and the attack path shown in Figure 12 give us a narrow difference final result 9995.78 EUR, if compared with final result from attacker profile 2, which is 9996.058 EUR. Parameters between attacker profile 2 and 6 are similar on budget (medium range) and time for launching the attacks if compared with the attacker profile 2. The only difference is the attacker skill, Medium (M) compared to Very High (V) on attacker skill on profile 2.
Table 7. Calculation result for Attacker Profile 6

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker 6</td>
<td>750</td>
<td>M</td>
<td>D</td>
<td>Gain : 9995.775390625 Max AS : {X7,X8,X16,X17,X22,X25,X26,X27, X30,X31,X35,X36}</td>
</tr>
</tbody>
</table>
Table 8. Calculation result for Attacker Profile 7

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker 7</td>
<td>50</td>
<td>L</td>
<td>D</td>
<td>Gain: 9246.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS: {X25,X26,X28,X30}</td>
</tr>
</tbody>
</table>

Calculation result shown in Table 8, gives us the maximum value 9246.75 EUR after 5 times running ApproxTree calculation program [17] is less compared to the previous attacker profiles that have stated before. The difference lies in budget (small amount) and also attacker skills (low), therefore the attack suite is only limited on the refinement of “usage policy violation”, which has the almost cheap attack path, from the entire attack tree.

Figure 13. Attack Path based on calculation result for Attacker profile 7

Table 9. Calculation result for Attacker Profile 8

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker 8</td>
<td>1500</td>
<td>L</td>
<td>HR</td>
<td>Gain: 9608.1513671875</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS: {X5,X6,X25,X26,X28,X30}</td>
</tr>
</tbody>
</table>
Attacker profile 8, shown in Table 9 and attack path Figure 14 has an opposite situation, which gives us elevation compared to the last attacker profile that we considered before with final result value 9608.15 EUR. The budget is considerably high with the same low level attack skills, and limited to hours (H) time to perform their attacks. Again the attacker has more options to choose because the budget resource is enabling attacker to finance attack suites that requires monetary funding like “impersonate PC technician” or “stole CC data”.

![Figure 14. Attack Path based on calculation result for Attacker profile 8](image)

### Table 10. Calculation result for Attacker Profile 10,11,13,14

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker 10</td>
<td>0</td>
<td>V</td>
<td>MT</td>
<td>Gain : 9788.4658203125</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS : {X25,X26,X27,X30,X31,X35}</td>
</tr>
<tr>
<td>Attacker 11</td>
<td>0</td>
<td>V</td>
<td>HR</td>
<td>Gain : 9788.4658203125</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS : {X25,X26,X27,X30,X31,X35}</td>
</tr>
<tr>
<td>Attacker 13</td>
<td>0</td>
<td>M</td>
<td>MT</td>
<td>Gain : 9788.4658203125</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS : {X25,X26,X27,X30,X31,X35}</td>
</tr>
<tr>
<td>Attacker 14</td>
<td>0</td>
<td>M</td>
<td>HR</td>
<td>Gain : 9788.4658203125</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS : {X25,X26,X27,X30,X31,X35}</td>
</tr>
</tbody>
</table>

Calculation result for attacker profiles 10,11,13,14 are shown in Table 10 and the attack path graph in Figure 15. We want to measure possibility that attacker with zero budget, but having various...
skills and time allocations to perform the attacks will affect the final result and attack path. Result shown in Table 10, it can be concluded that it is possible for attacker to gain quite significant final result, even higher if compared to result for attacker profile 8 which has high budget, but low level skill set. The reason why it is still possible to launch an attack with sophisticated result because there are attack suites that doesn’t require budget, but require some sets of attacker skills to perform the attack. For instance, the attack suites option such as “threat victim” or “impersonate CS supervisor” attacker needs to have specific individual weakness of victim or target and exploited it, and for “abuse reselling policy” or “abuse product”, attacker has to have a deep understanding on how the product policy role, how the product can be applied and seek the flaws in between the policies or products that can be played around.

Figure 15. Attack Path based on calculation result for Attacker profile 14

Calculation result shown in Table 11 and attack path graph in Figure 16, is the minimum result that we can get throughout our attacker profile variants. With zero budget and minimum level of skill set, attacker can still possibly result in an amount of profit which is significant without having lots of effort (monetary and skills wise). Therefore “abuse reselling policy” is usually chosen by the attacker to propagate the amount of profit that they received.
Table 11. Calculation result for Attacker Profile 16,17,18

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker 16</td>
<td>0</td>
<td>L</td>
<td>MT</td>
<td>Gain: 8000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS: {X30}</td>
</tr>
<tr>
<td>Attacker 17</td>
<td>0</td>
<td>L</td>
<td>HR</td>
<td>Gain: 8000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS: {X30}</td>
</tr>
<tr>
<td>Attacker 18</td>
<td>0</td>
<td>L</td>
<td>D</td>
<td>Gain: 8000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS: {X30}</td>
</tr>
</tbody>
</table>

On the contrary, with sets of attacker profiles with extreme zero budget, we provide attacker profile which has budget likewise the maximum gain, 10000 EUR with the highest possible attack skills, and deliberately days to perform the attacks. Thus resulted in Table 12 and attack path graph in Figure 17. Consider that those perfect conditions were intended to be compared to the existing attacker profile, has resulted the same exact result and attack path with attacker profile 2, which has only medium budget resource but the same skills and time resource. Final result of 9996.057 EUR and various choices of attack suites that an attacker can choose. Detail analysis will be given in next subsection.

Table 12. Calculation result for Attacker Profile 19

<table>
<thead>
<tr>
<th>Name</th>
<th>Budget</th>
<th>Skill</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker 19</td>
<td>10000</td>
<td>V</td>
<td>D</td>
<td>Gain: 9996.057</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max AS: {X7,X8,X16,X17,X21,X22,X25,X26,X27,X30,X31,X35,X36}</td>
</tr>
</tbody>
</table>
Result cannot be drawn from the attacker profiles 9, 12, 15, because the elementary attack parameter time has the bare minimum value measured in Minutes (MT) for attacker to be able to launch the attacks, therefore, no attack suites can be produced for these attacker profiles which have time allocation in Seconds (S).

Figure 17. Attack Path based on calculation result for Attacker profile 19

5.2 Analysis
We analyze the whole calculation result corresponding to the effect of 19 attacker profiles that we’ve presented in Table 1 towards final result from utilizing ApproxTree calculation program [17] presented in section 5.2.2. In section 5.2.1 we discuss in relation to attack tree attributes decoration estimation, what the estimation was based on, also elaborate attacker profiles attributes value choice, to highlight various type of actors involved onto action performing attack tree can give us new insight if attacker profile can derive maximum attack path with maximum final gain.

5.2.1 Attack Tree and Attacker Profile Attributes Decoration
Highlight of ApproxTree [17] is whether the analyzed systems are secure against rational gain-oriented attackers. It uses rationality concept from attacker point of view and economic reasoning to presents valuable conclusions about the security of the system. In addition, it provides insights
of the weakest link in the system which can be used by attacker, because this is most profitable attack vector to attack the system. Attacker profile onto ApproxTree [17], invalidates some nodes thus eliminates some sub-trees of the overall attack tree. Therefore, we get a reduced tree which describes possible attack path or attack steps for particular attacker.

The refinement on child nodes of case study attack tree were formed by “account take over”, “usage policy violation” and “CS agent compromised account”. “account take over” makes up majority of refinement compared to other child nodes, however leaf nodes from refinement “account take over” were considered expensive and requires skillful attacker to perform the attack suites especially if attacker only has a limited budget resource and low skill sets. Estimation of parameter values in leaf nodes were based on commonality of the similar attack, for example, “phishing victim”, “blackmail victim”, “quid pro quo”, “impersonate as PC technician”, “spear-phishing victim” attacker has to have skill sets to influence or convince victim to hand out the important piece of information from victim, or attacker can hire another person who will do the job. So, attacker needs to have specific skills or at least have budget to hire someone else to perform the attack.

We need to underline that estimation on “usage policy violation” has to be done based on an insight of internal policy of the particular digital products. Usage policy are usually displayed on website or when a customer signs up to the agreement of specific software. However, this kind of policy or rule of game sometimes requires deeper understanding what or how the product should work, person without knowledge of these policies cannot draw conclusions what can be flaws in the policies and the products itself. We believe each of the company who offers digital products has their own fraudulent detection system, which designed to prevent or limit the fraudulent activities that can cost loss revenue to the company. “abuse reselling policy” and “abuse product” is estimated on zero budget, and it doesn’t need a special skill set to exploited it, because presumably this type of actions attacker just need to affiliate with existing experienced fraudsters.

Estimations that were made for customer service agent are almost the same with individual or personal victim. However, it needs to be highlighted that customer service as part of the internal system of the company who sells the digital goods, moreover as a trend customer service nowadays are commonly made up by outsourcing in developing countries to keep up with the company
Budget efficiency. Compromising their account can lead to more destructive attacks towards the company.

Attacker profile parameters were already discussed in chapters 3 and 4, those attacker profile estimation values were designed to accommodate types of various attackers that are available in real world. However, we take only 19 samples on several attacker profiles who had budgets, varying from very high ($V = 10000$ EUR), high ($H = 2000$ EUR), Medium ($M = 500, 750$ EUR), and low ($L = 50, 0$ EUR), various skills or proficiency levels, and various times available for performing the attacks.

### 5.2.2 Calculation Result Analysis

Calculation result shown in Table 4 in Appendix, with the program ApproxTree are ran 5 times then we choose the maximum values that came out from those 5 batches of calculation results. Based on chapter 4, we can categorized the result according to the same value of final results as below:

1. Attacker profile 1, 3, 4, 5 with final result 9875.43 EUR
2. Attacker profile 2 and 19 with final result 9996.06 EUR
3. Attacker profile 6 with final result 9995.78 EUR
4. Attacker profile 7 with final result 9246.75 EUR
5. Attacker profile 8 with final result 9608.15 EUR
6. Attacker profile 10, 11, 13, 14 with final result 9788.47 EUR
7. Attacker profile 16, 17, 18 with final result 8000 EUR
8. Attacker profile 9, 12, 15 with no final result, no attack suite can be generated

There are leaf nodes that are always chosen in each of the attack path or steps that were generated during the calculation. “create new account”, “Report account blocking poses as legitimate user”, “Abuse reselling policy” is the easiest elementary attack that an attacker can perform regardless the attacker profile itself. The logic of calculation was based on attacker profile parameters included into input file. This will invalidate some of the leaf nodes which doesn’t comply with the attacker profile parameters.

Further observation point out that attacker profile 2 which had medium budget, has the same exact result if compared to the attacker profile 19 which has the maximum budget, both of the attacker
profiles have very high skill sets and also deliberately time to perform the attacks (days). In addition, attacker’s skill sets and time to perform an attack is very essential to determine the final result. For instance, the budget in attacker profile 1 and attacker profile 8, is nearly the same, and also the time they had is the same (hours), however attacker profile 1 has very high skills (V) and attacker profile 8 has low skill level, the result shows that attacker profile with very high skill resulted in more high final gain if compare to the low level skill attacker.

Time to perform the attack plays important role, therefore if attacker has only very limited time (seconds) then no attack steps or path can be generated because there aren’t leave nodes with parameter corresponding to perform attack in seconds.
Chapter 6
Conclusions and Future Research

This thesis observed and analyzed “attacker profile” as additional parameter for an attack tree, and also studies the effect of outcome produced by the ApproxTree calculation program [17] which has the base logic adopted from the Jürgenson – Willemsen parallel model [4] rationality concept on the part of attacker and economic reasoning to make meaningful conclusions about the security of the system. Use case study “Fraudulent Purchases of Digital Goods” was designed with the estimation of quantitative parameters for leaves of the tree in an actor-independent way as much as possible.

The effect of applying attacker profile to the attack tree will invalidate some nodes thus eliminates sub-trees from overall attack tree and provides possible attack steps for the particular attacker. Based on the calculation results that had been shown in Chapter 4 with various different types of attacker, the final outcome of calculation with various attacker profiles (AP 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14, 16, 17, 18) is not significantly different from calculation results preformed without attacker profile (AP 19). The calculation results on various attacker profiles using ApproxTree [17] ranged only 0 % until maximum 20 % against final gain (10000 EUR).

There are several indication that can be withdrawn from this particular calculation result from the initial case study: First, because result from different variety of attacker profiles doesn’t seem to have any effect the final result, it is important to highlight that different sets of attacker profile can give the same exact outcome, including the same attack path. This is caused by the parameters of elementary attacks (leaf nodes) corresponding the set of attacker profile parameters. For example in Table 5, Attacker 1 and attacker 5 who clearly have huge gap on its budget, skills and time allocations still can produce the same exact attack suites, because the parameters of attack tree chosen are in the range of attacker profile parameters, so calculation resulted on the same value as no difference between the 2 attacker profile. Attacker profile 19 represents condition without attacker profile, however, the result is still the same with attacker profile 2 in Table 6. This goes with the same analogy with correlation between attacker profile parameters and leaf nodes parameters, thus enabling middle budget attacker with very high skill and without time constraint to produce the same results.
Second conclusion is related to attacker parameters and parameters value effect of leaf nodes on calculation using ApproxTree [17]. Table 10 and 11, provided with zero budget attacker profile with various skills and time allocations, but Table 10 shows slightly higher result compare to zero budget low skill. Thus, shows that constraints in budget enable low skilled attacker sufficient time to exploit the system through elementary attacks which have zero budget and easy level of difficulty to be executed, attacker might do trial and error during performing the attack and can achieve nearly the same result if compared to middle budget and high skilled attacker.

Third conclusion is to point out a correct notion for attacker regardless their budgets and their skills, if time allocations is only in seconds, attacker will not be able to perform elementary attacks. In real conditions there are no shortcut or easy way to launch an attack, therefore time resources or allocation for attacker is deemed to be important parameter to effect the outcome. Based on calculation results, it can be concluded that budget is not important as other parameters such as attacker skills and time allocation to perform attack. The higher skills and time allocation available, the higher possibility for attacker to succeed, even though attackers have minimum budget, there won’t be any hacking technique constraint and ability to create his/her own tool to exploit the weak security link in the system.

Furthermore, final conclusion need to be drawn here, despite of no significant difference on result between upper limit and lower limit in terms of final gain that attacker achieve, we are confident that enclosing attacker profile onto attack tree is a useful concept for quantitative security assessment based on attack tree methodology.

Quantitative security assessment analysis, attack tree in particular. Without attacker profile gives us insight regarding if the system is sufficiently secure against all possible attack scenarios, this means that the attacker is overpowered. However in real life condition attackers have certain constraints and cannot deliberately perform all possible attack scenarios towards the targeted system. Result of analysis without attacker profiles might result in “False Negative” or underestimating security of the measured system. It doesn’t mean that the measured system is not secure in reality, but might be insecure against overpowered attacker which we consider, and it still be secure against real world’s less powerful attacks. In contrary, result of analysis with certain attacker profiles might result in “False Positive” or overestimating security of the measured system. This means that all possible attacker profiles have been considered in analysis, and the
result of the analysis is that the system is secure, but in real life the system maybe insecure if the profiles were not estimated correctly. The result gap between analysis without attacker profile (minimum value 0 %) and with attacker profile (maximum value 20 %) based on case study Chapter 4, can be translated as 20% is the required investment that companies or owner of measured system to upgrades their system towards real or near to ideal preferred system.

As for future research, it is needed to use more various comprehensive data according toward actual conditions as based of parameterizations of the leaf nodes when composing the attack tree and consider to add more parameters to attacker profiles, for instance: confidence level, motivation, or even the quantity of attacker (collective or individual). We should definitely overview and improve the algorithm used to calculate the attacker’s outcome for delivering more various and realistic approximate attacker’s outcome value.

In addition, ongoing research for attacker profiling is implementation of attacker profile in compliance with FAIR framework [80,81], Factor analysis of information risk or known by abbreviation FAIR is a taxonomy of the factors that contribute to risk and how they affect each other. The main contribution of FAIR in future research is to elaborate possible studies on inter-relations between attack tree parameters and attacker profile parameters, also provides possible type of interactions between these parameters. For example, a more skilled attacker will have higher probability of success than the less skilled one, given the same amount of time. But if a skilled attacker has much less time than the less skilled one - on the contrary, his probability of success may be even less than the less skilled attacker's.
**Resümee**


References


Fraudulent Purchases of Digital Goods
Account Take Over
Blackmail Victim
Social Engineer Victim to Reveal Account Data
Quid pro quo
Phishing Victim
Infect Victim PC with Malware
Get Access to Victim PC
Break into Premises
Steal PC
Pretexting
Impersonate PC Technician
Insert Malicious USB
Spear-phishing victim
Bait victim to Malicious Website
Web Obfuscation
Shoulder Surfing
Learn ID & Password
Guess ID
Brute Force Password
Dictionary Attack
Get CC Data
Threat victim to reveal CC data
Stole CC data
Buy CC data from Underground Hacker's Forums
Sniffing Network Traffic
Wireless Tapping
Wired Tapping
Get Access to ISP Office
Apply for Job in ISP Company
Bribe ISP Employee
Social Engineer ISP Employee
Tailgating
Phishing ISP Employee
Baiting ISP Employee
ISP Traffic Tapping
Remote Tapping
Hack ISP Gateway
Install Sniffer
Bypass Usage Policy
Abuse Reselling Policy
Abuse Product
CS Agents Compromised Account
Bribe CS Agents
Identify Corrupt CS Agents
Bribe target
Social Engineer CS Agents
Phising CS Agents
Impersonate CS Supervisor
Baiting CS Agents
Threat CS Agents
Blackmailed CS Agents
APPENDIX 2

TABLE 4. OVERALL RESULTS CALCULATION APPROXTREE PROGRAM
### Table 4. Overall Result from ApproxTree Calculation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st Batch</td>
<td>2nd Batch</td>
<td>3rd Batch</td>
<td>4th Batch</td>
<td>5th Batch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max AS : {X7,X8,X16,X17,X21,X22,X25,X26,X27,X30,X31,X35,X36}</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : {X25,X26,X27,X28,X30,X31,X35}</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : {X25,X26,X27,X30,X31,X35}</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : {X30}</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
<td>Max AS : None Attack suite can be Generate</td>
</tr>
<tr>
<td>MT</td>
<td>HR</td>
<td>S</td>
<td>V</td>
<td>D</td>
<td>MT</td>
<td>HR</td>
<td>S</td>
<td>V</td>
<td>D</td>
<td>MT</td>
<td>HR</td>
<td>S</td>
<td>V</td>
<td>D</td>
<td>MT</td>
<td>HR</td>
<td>S</td>
<td>V</td>
<td>D</td>
</tr>
</tbody>
</table>

**Note:** The table details the results from the ApproxTree calculation for various attackers across different batches, with specific gains and maximum attack suites indicated for each.