Calculating Adversarial Risk from Attack Trees: Control Strength and Probabilistic Attackers

Wolter Pieters[1,2](✉) and Mohsen Davarynejad[1,3]

[1] Technology Policy and Management, ICT, Delft University of Technology, Delft, The Netherlands
w.pieters@tudelft.nl, mohsen@davarynejad.com

[2] EEMCS, Services, Cybersecurity and Safety, University of Twente, Enschede, The Netherlands

[3] Department of Radiation Oncology, Erasmus Medical Center, Daniel den Hoed Cancer Center, Rotterdam, The Netherlands

Abstract. Attack trees are a well-known formalism for quantitative analysis of cyber attacks consisting of multiple steps and alternative paths. It is possible to derive properties of the overall attacks from properties of individual steps, such as cost for the attacker and probability of success. However, in existing formalisms, such properties are considered independent. For example, investing more in an attack step would not increase the probability of success. As this seems counterintuitive, we introduce a framework for reasoning about attack trees based on the notion of control strength, annotating nodes with a function from attacker investment to probability of success. Calculation rules on such trees are defined to enable analysis of optimal attacker investment. Our second result consists of the translation of optimal attacker investment into the associated adversarial risk, yielding what we call adversarial risk trees. The third result is the introduction of probabilistic attacker strategies, based on the fitness (utility) of available scenarios. Together these contributions improve the possibilities for using attack trees in adversarial risk analysis.

Keywords: Adversarial risk analysis · Attack trees · Attacker models · Control strength · Fitness functions · Security metrics · Simulation

1 Introduction

Attack trees [8,9,15] are a well-known formalism for analysing cyber attacks consisting of multiple steps and alternative paths. It is possible to derive properties of the overall attacks from properties of individual steps, such as cost for the attacker, probability of success, and probability of detection. In existing formalisms, such properties are considered independent. For example, investing more in an attack step would not increase the probability of success. This even holds for more complicated schemes in which multiple parameters are considered simultaneously [4].

J. Garcia-Alfaro et al. (Eds.): DPM/SETOP/QASA 2014, LNCS 8872, pp. 201–215, 2015.
DOI: 10.1007/978-3-319-17016-9_13

Although such approaches have definitely shown their value in both theoretical and practical respect, there are several issues. Firstly, assuming that an attack step always costs the same in any situation seems counterintuitive. An attacker who wants to be really sure that a particular step succeeds may invest more time and/or money, thereby increasing the likelihood of success (and maybe reducing the likelihood of detection). Secondly, analysis tools are available which work precisely with a relation between investment (time) and likelihood of success [1]. To enable the attack tree paradigm to take full advantage of such methods, they need to support suitable annotations (i.e. dependent parameters).

To address these issues, this paper proposes a framework for reasoning about attack trees based on the notion of control strength, which is a function from attacker investment to probability of success. Calculation rules on such trees are defined to enable analysis of optimal attacker investment, in the context of a two-step game where the attacker can choose the optimal attack after the defender has placed his controls. The key application of this approach is in adversarial risk assessment. Whereas traditional attack trees were not directly connected to the notion of risk, our approach enables their use in calculating adversarial risk in terms of threat, vulnerability, and impact, where vulnerability describes the relation between attacker investment and probability of success, and threat describes the optimal attacker strategy based on vulnerability and impact.

In Sect. 2, we provide definitions for parameters of interest and analysis of attack trees with dependent parameters, based on optimal attacker investment. Section 3 shows an example including simulations. In Sect. 4, we illustrate how to factor in time, to calculate risk based on optimal attacker investment. In Sect. 5, we extend the approach with probabilistic attackers, assuming that limited knowledge and limited rationality will lead attackers to not always choosing the optimal scenario, creating a different risk picture for the defender. In Sect. 6, we discuss related work, and we conclude in Sect. 7.

2 Definitions

2.1 Preliminaries

Factor Analysis of Information Risk (FAIR). To define our concepts (the same as in [13,14]), we use the risk definitions provided by The Open Group [16]. In this taxonomy, risk-related variables are defined starting from the notions of assets and threat agents acting against these assets, potentially causing damage. A threat event occurs when a threat agent acts against an asset, and a loss event occurs when this causes damage. For example, a storm may occur at the location of a power line (threat event), and this may or may not damage the power line (loss event).

Like many other approaches, The Open Group distinguishes between what they call Loss Event Frequency (LEF) and Probable Loss Magnitude (PLM). The former represents the expected number of loss events of a particular type per unit of time (often referred to as likelihood), and the latter represents the expected damage per loss event of that type (often referred to as impact).

Risk can be seen as expected damage due to a certain type of loss event within a given time frame, and it can then be calculated as LEF · PLM.

Within LEF and PLM, The Open Group makes further distinctions. We will not discuss PLM here, but focus on LEF. First of all, the Loss Event Frequency can be separated in Threat Event Frequency (TEF) and Vulnerability (V). TEF denotes the expected frequency of occurrence of a particular threat (seen as a threat agent acting against an asset; a storm at the location of a power line), and V specifies the likelihood of the threat inflicting damage upon the asset. The value for LEF can then be calculated as TEF · V. The Open Group defines the Vulnerability V based on Threat Capability (TC) and Control Strength (CS). In this definition, TC denotes some ability measure of the threat agent, and CS a resistance (or difficulty of passing) estimate of the control. We have discussed this relation in detail in [14]. Note that the term vulnerability is used as probability of success here, not as a software bug causing a security weakness. To avoid ambiguity, we will only use the term probability of success in this paper.

Attack Trees. Attack trees [15] describe attacks by means of a tree structure, in which attacker goals are refined by means of AND-nodes (all subgoals have to be achieved) and OR-nodes (only a single subgoal has to be achieved). In the end, attack trees represent a set of possible attack paths [9]. Figure 1 shows a simple attack tree for obtaining secret data. The tree consists of four leaves. The left branch is an AND-node; the other two non-leaf nodes are OR-nodes.

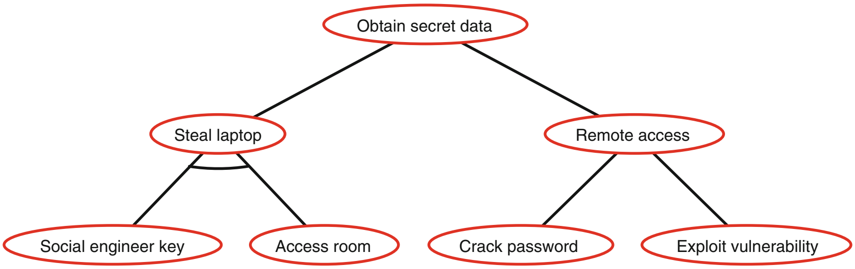

Fig. 1. An attack tree for obtaining secret data by laptop theft or remote access in ADTool [7]. The bottom left non-leaf node is an AND-node, the other non-leaf nodes are OR-nodes.

Attack trees can be annotated with all kinds of values, such as probability of success, cost for the attacker, required time, etc. Many calculations are possible on such trees, from simple bottom-up value aggregation from leaves to root, to complicated calculations made feasible by genetic algorithms. An extensive literature review of work to date on attack trees is provided in [8]. Some dependency between variables is taken into account in [4], in the sense that the probability of detection depends on whether the attack step succeeded or failed. However, the cost of an attack step is still fixed.

2.2 Contributions

Our innovation lies in the explicit use of control strength and threat capability in the analysis of attack trees. This separates system properties and attacker properties. In addition, we distinguish between static attacker properties (skill) and dynamic attacker properties (investment). The former are pre-defined and constant over the duration of the attack; the latter can be strategically chosen by the attacker.

In particular, we consider control strength (also called difficulty) as a function from threat capability to probability of success. By defining difficulty in this way, we enable the use of separate, explicit attacker profiles, that can contain static properties (skill) and rules for dynamic properties (investment strategy). In this paper, we assume only a single investment strategy, namely the (rational) one that maximises utility for the attacker. However, we do take account probabilistic deviations from optimal selection of scenarios (Sect. 5). The question on how to define the difficulty function has been treated elsewhere [14], and we will not repeat this issue here. The definition of difficulty as a function also implies that attacker investment (cost) and probability of success cannot be treated as independent properties, like in existing formalisms. This adaptation seems intuitive, as the more an attacker invests in trying to get in, the more likely he will be able to succeed. Thus, the existing assumption of fixed costs and a fixed probability of success for an attack step is lifted.

To be able to calculate attacker utility, we assume there is a certain value for the attacker associated with the goal represented in the root node of the attack tree. For example, achieving the root goal would provide a utility of € 1,000 to the attacker. We will not address the question how to define this value in the present work. The model presented here is a *parallel model* [4] in which all atomic attacks $x_1, \ldots, x_n$ take place simultaneously; e.g. the adversary chooses a subset of atomic attacks and executes them in parallel independent of success or failure of some of attack steps.

2.3 New Definitions

Definition 1. *The* probability of success *of an attack step or composite attack is a value in the range [0,1], indicating how likely the attack (step) is to succeed when executed. The probability of success can also be interpreted as an expectation value of the success variable, when failure is 0 and success is 1.*

Note that this is different from what is generally referred to as likelihood in risk assessment, as that likelihood refers to probability (frequency) of occurrence, not probability of success.

Definition 2. *A* control strength function *is a function $c : T \rightarrow [0,1]$, indicating the relation between threat capability and probability of success. T can be any partially ordered set suitable for representing capability (in its simplest form {Low, Medium, High}, but could also be money). It is assumed that c is monotonic: a higher threat capability will lead to a higher or equal probability of success.*

Definition 3. *A* threat capability function *is a function $t : S \times I \rightarrow T$, indicating the relation between attacker skill, attacker investment, and threat capability. S could again be for example {Low, Medium, High}. I is typically expressed in terms of money. It is assumed that t is monotonic in both parameters: a higher skill or a higher investment will lead to a higher or equal threat capability.*

Definition 4. *An* investment function *is a function $f : I \rightarrow [0, 1]$, indicating the relation between attacker investment and probability of success. It is assumed that f is monotonic: a higher investment will lead to a higher or equal probability of success.*

An investment function f can be expressed in terms of a control strength function c, a threat capability function t, and an attacker skill s as $f(i) = c(t(s, i))$. If both c and t are monotonic, f will be monotonic as well. Nodes in attack trees can now be annotated with control strength functions rather than simple values. For non-leaf nodes, the functions represent the results for the associated subtree. If no explicit attacker profiles (e.g. skill) are considered, an annotation with investment functions suffices.

2.4 Analysis

The analysis is based on the formalisation of attack trees by Mauw and Oostdijk [9], in which the semantics of an attack tree is the corresponding set of attacks C, which are multisets of attack steps. It determines what constitutes the optimal investment for the attacker (maximum utility), by finding out how to distribute resources as investments over attack steps. To this end, we first need to determine how the investment function of AND- and OR-nodes can be derived from those of their children. The calculation rule for the investment function f of an AND-node with children x_1 and x_2 with investment functions f_1 and f_2 is:

$$f(j) = max\{f_1(i_1)f_2(i_2) \mid i_1 + i_2 \leq j\} \tag{1}$$

Note that, under the assumption that f_1 and f_2 are monotonic, the maximum will always occur when $i_1 + i_2 = j$. The calculation rule for the investment function f of an OR-node with children investment functions f_1 and f_2) is:

$$f(j) = max\{f_1(i_1) + f_2(i_2) - f_1(i_1)f_2(i_2) \mid i_1 + i_2 \leq j\} \tag{2}$$

In this equation, it is assumed that the attacker can invest in multiple branches of the OR-split in order to maximise his probability of success. In practice, the attacker may first try one branch and base his further investment upon the result. We do not discuss such “sequential OR-nodes” in this paper, but they would be relevant for future studies.

In addition to combining the investment functions of the children into one, the analysis would need to keep track of the distribution of investment over the subtrees associated with the maximum probability of success (the argmax). As can be seen from the definitions, the functions will get increasingly complicated when moving towards the root node (at least in the general case). For each branch of the tree, the function definition could be split in two separate domain intervals. In practice, one will not do these calculations before a specific question has been asked on the tree. In particular, the question of the optimal attack from the attacker perspective is relevant here.

In this paper, we assume that an attacker only gains utility if he achieves the root goal. The optimal balance between investment and probability of success depends on how much utility the root node provides. The higher this utility, the more important the probability of success becomes, compared to the investment.

3 Examples and Simulations

In this section we provide a number of examples to illustrate the theoretical properties of investment trees. The examples provided here are small; they are meant for illustrative purposes. This means that we can exhaustively enumerate the possible attack paths and their utilities, which is useful for explaining the intuitions. In these examples the cumulative distribution functions (CDFs) of logistic distributions are adopted to mathematically represent the probability of success as a function of investment (according to Definition 4). There are two reasons for choosing logistic distributions: (1) logistic distributions provide a strong analogy with difficulty metrics in other domains [14], and (2) the logistic distribution reflects the fact that with little investment, the probability of success is low, whereas there is a certain critical point around which the probability of success increases rapidly with higher investment. There are three parameters of interest: the mean μ of the distribution, the scale s, and the maximum success probability c. The latter reflects the fact that a probability of success of almost 1 is not always achievable, not even with nearly infinite resources.

Figure 2 shows an example of such a function with $\mu = 33$, and $s = 0.15$ (blue dash curve) or $s = 0.44$ (dot-dash red curve). The units are not specified here, but investment can be thought of as units of time or money the attacker invests in an attack step. In all the simulation results provided below the assumption is that the attacker is rational, so s/he launches multi-stage attacks to achieve his/her goals. Other distributions like Rayleigh CDF might be also suitable for particular cases, but the essentials remain the same. In an ideal scenario these functions can be estimated from empirical evidence like penetration tests [2].

Example 1. Suppose that the attacker goal is to steal a laptop, which will yield a gain (positive utility) of 1000 for the attacker, and a damage (negative utility) of 5000 to the defender (e.g. replacement plus data loss). The attacker can invest in two branches of an AND-split. The first branch x_1 has an investment function f with a logistic CDF form with $\mu_1 = 33$ and $s_1 = 0.15$. The second branch x_2 has exactly the same investment function as x_1 with the exception of $s_2 = 0.44$. Figure 2 shows the investment function and Fig. 3a the profit landscape. Each black asterisk represents the optimal investment in each attack step when the sum of resources j is limited. In this example j is set at 0 and increases to 300 with step size of 2.

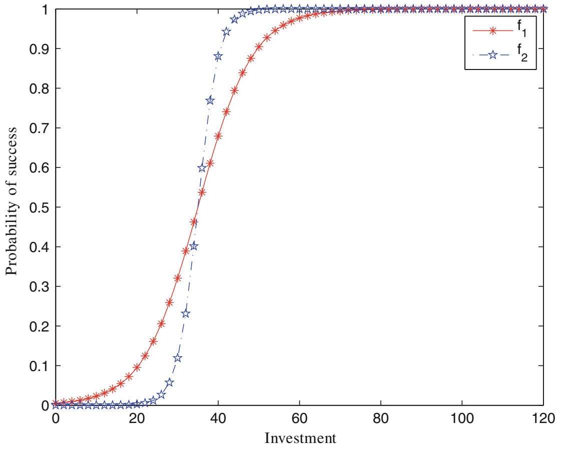

Fig. 2. Investment function f_1 with $\mu_x = 33$, $s_x = 0.15$, and f_2 with $\mu_y = 33$, $s_y = 0.44$ (Color figure online)

Example 2. Suppose that the attacker goal is to steal a laptop, where she can invest in two branches of an OR-split. The first branch has an investment function f_1 similar to that of the first branch of Example 1 and the second branch x_2 has exactly the same investment function as that of the second branch of Example 1. Then the profit landscape is shown in Fig. 3b. Here again each black asterisk represents the optimal investment in each attack step when the sum of resources j is limited. In this example j is set at 0 and increases to 300 with step size of 2. The maximum resources the attacker can spend is the same as in the first example. Because the attacker can choose between two alternatives with different investment functions, the optimum jumps from one alternative to the other depending on the amount the attacker can invest.

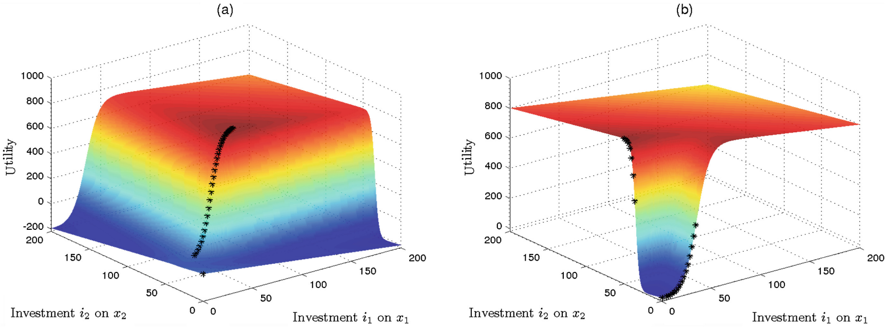

Fig. 3. Profit landscape for (a) the AND-node example, (b) the OR-node example.

Example 3. Now consider the attack tree from Fig. 1, with leaf node investments labelled i_1, i_2, i_3, i_4, from left to right. In this example f_1 and f_2 (left branch) are the same as what we have described in previous examples. The μ_3 and μ_4 (right branch) are both set at 75 and s_3 and s_4 are set at 0.15 and 0.4 respectively (Fig. 4). The optimal investment on each attack step for a number of values of j is reported in Table 1. When the attacker's maximum investment j is less than 56, then the empty strategy (i.e. not playing at all) is optimal.

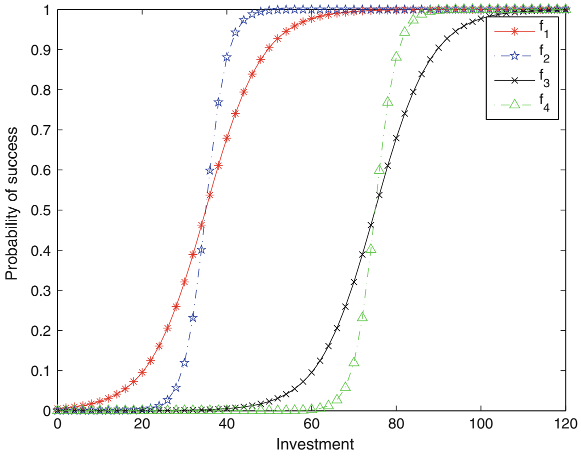

Fig. 4. Investment functions for the complete example attack tree.

These examples show how to use control strength in attack tree calculations, optimising the relation between investment and probability of success from the point of view of the attacker. However, this does not directly provide information about adversarial risk.

4 From Optimal Investment to Risk

Many papers on security risk assessment end with the analysis of optimal investments, without considering time. We take one step further here, and try to align security risk assessment with safety risk assessment by considering attack frequencies, based on a profile of the attacker in terms of available resources and investment strategy. As in the previous sections, we choose the Factor Analysis of Information Risk taxonomy [16] as the risk assessment framework. We choose this taxonomy because it explicitly relates threat capability (investment), control strength, and probability of success, similar to our extended attack trees. In this framework, vulnerability is synonymous to probability of success, and this vulnerability is dependent on both threat capability and control strength. From the point of view of the attacker, the control strength (strength of the defense)

Table 1. Optimal investment on each attack step when j is incremented from 50 to 100 with step size of 4.

j	Optimal investment on each attack step				Utility	Total investment
	i_1	i_2	i_3	i_4		
50	0	0	0	0	0.013011	0
54	0	0	0	0	0.013011	0
58	21.2594	36.7406	0	0	17.3807	58
62	24.9633	37.0367	0	0	63.8835	62
66	0	0	66	0	139.8704	66
70	0	0	70	0	250.8213	70
74	0	0	74	0	388.5702	74
78	0	0	0	78	690.5278	78
82	0	0	0	82	860.6766	82
86	0	0	0	86	901.8717	86
90	0	0	0	89.9661	907.5276	89.9661
94	0	0	0	89.9661	907.5276	89.9661
98	0	0	0	89.9661	907.5276	89.9661

is fixed, and vulnerability can thus be expressed as a function from investment (in threat capability) to probability of success (vulnerability).

As risk is expressed as threat event frequency (likelihood) times vulnerability times impact, we need to address the missing item, which is the frequency. To this end, we need to extend the analysis from a single point in time decision by the attacker to longitudinal, by limiting the attacker income (and thereby his resources) as a function of (continuous) time. In this paper, we choose the discrete event model from [13] for the analysis of risk, based on attack trees endowed with control strength annotations for optimal investment analysis. In the discrete event model, attackers save resources and attack with the accumulated resources at a single point in time. After the attack, the damage is assumed to be repaired, the attacker resources are reset to zero, and the same process will be repeated. Whereas [13] only discusses atomic attacks, we apply the analysis to attack trees here. In addition, we extend the analysis with non-zero-sum situations, as well as probabilistic attacker strategies. The mapping from optimal investment to risk, via time, proceeds as follows [13]. Attackers can, at each point in time, choose to launch an attack with the resources they have built up until that point ($R(t)$). Attackers will also have a skill level s, and the threat magnitude m is a function of the skill and the available resources:

$$m(t) = f(s, R(t)) \tag{3}$$

Attackers will not attack, but rather wait and save resources, if they can gain higher expected average utility by launching an attack with more resources later.

If they cannot improve their average utility by waiting, they will execute the scenario with the highest expected utility given their current resources.

The vulnerability (probability of success) V of an attack step/scenario c depends on the threat magnitude m. The expected utility $U_c(m)$ for each threat capability level m and for each attack scenario c is:

$$U_c^A(m) = V_c(m) \cdot G_c^A \tag{4}$$

where G_c^A is the utility (gain) for the attacker upon *success* of scenario c.

The maximum utility for a given threat capability level m is specified by

$$\hat{U}^A(m) = \max_{c \in C} U_c^A(m) \tag{5}$$

The optimal scenario to execute is then $\operatorname*{argmax}_{c \in C} U_c^A(m)$.[1]

We can therefore calculate the maximum expected utility at each point in time, $\hat{U}^A(t) = \hat{U}^A(m(t))$, and also the maximum average utility over the elapsed time, which we denote S for *success*, $\hat{S}^A(t) = \hat{U}^A(m(t))/t$. Assuming an attacker who wants to maximise his average gain per unit of time, the attacker will thus attack at the time $\hat{t}$ when $\hat{S}^A(t)$ reaches its maximum. The scenario that will be executed is

$$\hat{c} = \operatorname*{argmax}_{c \in C} U_c^A(m(\hat{t})) \tag{6}$$

Assuming instant repair, the expected threat event frequency can be determined as $h_{\hat{c}} = 1/\hat{t}$. For all other scenarios, the threat event frequency is zero. The loss event frequency for scenario $\hat{c}$ is $\lambda_{\hat{c}} = V_{\hat{c}}/\hat{t}$. From the loss event frequency, we can calculate the risk (negative success) S^D as annual loss expectancy, by filling in the utility for the defender rather than the attacker of a successful scenario c. Note that both $U_{\hat{c}}^D$ and S^D are negative.

$$S^D = h_{\hat{c}} \cdot U_{\hat{c}}^D \tag{7}$$

Example 4. Consider the attack tree of Example 3. We assume that the skill level is irrelevant here, and we therefore assume that $m(t) = R(t)$. The attacker has income density function $\frac{dR}{dt} = 1$, i.e. the attacker will earn 1 resource unit per unit of time, or $m = t$. Up to $t = 1$, the attacker will thus be able to invest 1 unit. In Fig. 5, the resulting optimal utility function $\hat{S}^A(t)$ is shown.

Note that in the discrete model, it is required that $V(0) = 0$. Otherwise, the expected risk (damage per unit of time) for very small t would be very high (up to infinite with t approaching 0), and the attacker would simply launch loads of "mini-attacks" with almost zero effort. For logistic vulnerability models where V is not zero as t approaching 0 additional considerations are required. In this study, this property of logistic vulnerability models is handled by setting the step size of change in t to a value bigger than 1.

[1] Note that picking argmax may involve nondeterministic choice if multiple arguments produce the maximum. This is one of the issues that the probabilistic attackers in this paper (Sect. 5) help to solve.

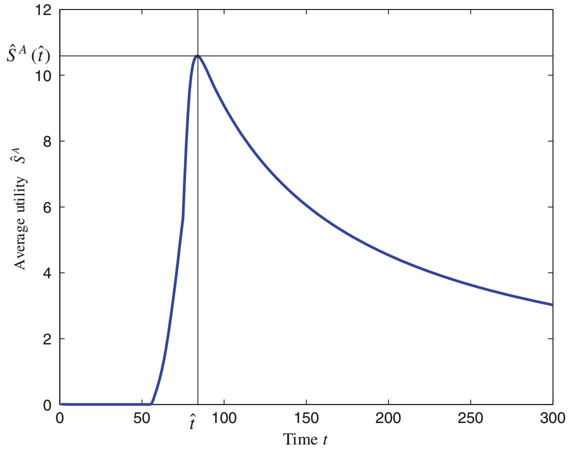

Fig. 5. A simulation of attacker success (utility per unit of time) as a function of attack time for the laptop theft scenario (Example 3). Initially, it does not make sense to invest at all. From $t = 58$, utility increases rapidly (see also Table 1), but success (utility per unit of time) starts to decline when marginal utility decreases.

5 Probabilistic Attackers

When evaluating the effect of countermeasures, an optimising attacker model implies that defenses that are not on the critical (optimal) path will have no effect. This is not very realistic. In reality, attackers have only limited information and need not be fully rational. To account for attackers not always choosing the optimal attack, we introduce probabilistic attacker strategies. In such a strategy, the probability of selecting a particular scenario is based on the fitness of that scenario, expressed as the utility it provides to the attacker per unit of time. For now, we assume that the attacker will always execute a chosen scenario at the optimal point in time. This is still a worst-case approximation, but it does not matter that much for defender decisions on effectiveness of countermeasures. Our probabilistic attacker is *not* meant to model a strategic attacker that knows everything, but chooses less optimal scenarios to evade defender actions. It is only meant to represent uncertainty on the part of the attacker.

Our proposal for assigning selection probability P_c to attack scenario c has its roots in genetic algorithms. The higher the utility of an attack scenario S_c^A, the higher its probability of being selected. This represents an attacker optimising his selection while possessing limited information about the fitness of the scenarios. In order to differentiate between levels of knowledge and understanding of the system, we use Boltzmann selection [10], which defines selection probabilities in the form of Boltzmann canonical distributions. It has been shown that the Boltzmann distribution can be derived from maximizing the Shannon entropy under proper constraints [11]. The selection probability is assigned as follows:

$$P_c = \frac{exp(-T/f_c)}{\sum\limits_{c \in C} exp(-T/f_c)} \tag{8}$$

where $T \in \mathbb{R}^+$ represents the level of knowledge and understanding of attacker concerning the system. The higher T, the higher the knowledge of the attacker, which results in better identification of high utility attack scenarios. In contrast, lower T represents a poor understanding of the system for the attacker, which results in poor distinction between high- and low-profit attack scenarios. This poor understanding results in higher uncertainty and insignificant difference in probability of selection of attack scenarios. In Eq. (8) f_c is described as:

$$f_c = \frac{S_c^A}{\sum\limits_{c \in C} S_c^A} \cdot |C| \tag{9}$$

Example 5. We take the attack tree presented in Example 3 (Fig. 1). This attack tree has three attack scenarios: $\{x_1, x_2\}$, $\{x_3\}$, and $\{x_4\}$. The optimal attack time $\hat{t}$ for each of these attack scenarios and their respective success values at optimal attack time $S_c^A(\hat{t})$ and $S_c^D(\hat{t})$ are reported in Table 2.

Table 2. Attack scenario selection probability.

c	$\hat{t}$	$S_c^A(\hat{t})$	$S_c^D(\hat{t})$	T, attacker knowledge level			
				$T = 0.1$	$T = 1$	$T = 10$	$T = 100$
$\{x_4\}$	84	10.59	−56.94	0.3375	0.3761	0.7512	1.0000
$\{x_3\}$	92	09.08	−49.41	0.3326	0.3247	0.1728	0.0000
$\{x_1, x_2\}$	96	08.04	−46.03	0.3299	0.2991	0.0760	0.0000
Attacker $\bar{S}^A$				9.3168	9.3944	10.1260	10.5881
Defender $\bar{S}^D$				−50.5810	−50.9381	−54.3882	−56.9407

The analysis proceeds as follows:

1. Calculate fitness (attacker success S_c^A) for all attack scenarios, assuming optimal time of attack $\hat{t}$; also calculate defender success (risk);
2. Assign probabilities P_c to scenarios based on their fitness in accordance with Eq. (8);
3. Calculate expected attacker success and defender succeess by weighted average.

The calculation of the weighted average is:

$$\bar{S}^A = \frac{\sum\limits_{c \in C} P_c U_c^A}{\sum\limits_{c \in C} P_c t_c} \tag{10}$$

The assumption is that the attacker chooses a scenario first based on the assigned probabilities, and then spends the associated time executing that scenario. This means that the time spent on a particular scenario is dependent on both the probability of selecting that scenario, and the time taken to execute the scenario. The expected utility of the whole strategy (all scenarios, weighted by probability), can then be divided by the expected time spent to find the average utility per unit of time. Replacing U_c^A with U_c^D, the same equation can be used to calculate the expected (negative) utility for the defender.

6 Related Work

Many of the annotations on attack trees, including time, cost and probability, are dependent both on properties of the system (e.g. resistance) and properties of the attacker (e.g. skill). In the TREsPASS project, we are looking for attacker-independent metrics, such that different attacker profiles can be used on the same attack tree. The idea of using control strength as a metric stems from the Factor Analysis of Information Risk framework, included in the Risk Taxonomy of The Open Group [16]. This idea was further developed in [2,14]. In this paper, we define control strength as function from attacker investment to probability of success.

In [1], a time-dependent analysis of attack trees is provided, relating time and success probability. However, investment (cost) is not considered, and the results are not linked to risk. The approach is also not focused on attacker decisions (in which step to invest). Because an exponential distribution is assumed, *the best choice never depends on the available time*, as the corresponding time-probability (cumulative probability) curves never intersect. For an OR-node, simply the fastest subtree is chosen; for an AND-node, both are started in parallel; for the additional sequential SEQ-node, the first subtree has to succeed first.

Several authors have considered return on attack as a security metric [3,6]. The lower the return on attack, the more secure the system. The return on attack complements the return on security investment from the defender's point of view. Several game-theoretic approaches have tried to relate the two. In this paper, we used the minimax approach suggested in [5], which optimises the defender investments under the assumption that the attacker will optimise his return on attack in the next step.

7 Conclusions

In this paper, we provided a new type of analysis on attack trees, namely the analysis of optimal attacker investment, under the assumption that the investment in an attack step influences the probability of success for that step. The formalisation in terms of control strength (difficulty) as a function from threat capability to probability of success was inspired by the Risk Taxonomy of The Open Group [16]. This constitutes an innovation beyond traditional formalisms, which considered attacker cost and probability of success as independent parameters. In addition, we showed how the analysis of optimal investment can be used to define adversarial risk in the context of attack trees, by limiting attacker resources in time (income). Finally, we showed how probabilistic attacker models, based on fitness evaluation of different scenarios, can improve the risk analysis, in particular when it comes to evaluation of countermeasures. The approach can be used to enhance existing security metrics, such as for example weakest link [12], adversarial risk, and return on security investment.

In this paper, we assume that the attacker makes his investment decisions upfront. In future work, we will investigate how the framework changes if the attacker can adapt his investment decisions on the fly, and how cooperation between multiple attackers influences the results. Also, we may consider the situation where other nodes than the root node would provide positive utility to the attacker upon success, and the situation in which the attacker invests his gain in new attacks. In addition, the probability of detection and punishment may be included next to the probability of success. Finally, case studies could provide further insights into the behaviour of the simulations.

Acknowledgements. The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007–2013) under grant agreement number ICT-318003 (TREsPASS). This publication reflects only the authors' views and the Union is not liable for any use that may be made of the information contained herein.

References

1. Arnold, F., Hermanns, H., Pulungan, R., Stoelinga, M.: Time-dependent analysis of attacks. In: Abadi, M., Kremer, S. (eds.) POST 2014 (ETAPS 2014). LNCS, vol. 8414, pp. 285–305. Springer, Heidelberg (2014)
2. Arnold, F., Pieters, W., Stoelinga, M.I.A.: Quantitative penetration testing with item response theory. In: 2013 Proceedings of Information Assurance and Security (IAS). IEEE (2013)
3. Bistarelli, S., Fioravanti, F., Peretti, P.: Defense trees for economic evaluation of security investments. In: 2006 The First International Conference on Availability, Reliability and Security, ARES 2006 (2006)
4. Buldas, A., Laud, P., Priisalu, J., Saarepera, M., Willemson, J.: Rational choice of security measures via multi-parameter attack trees. In: López, J. (ed.) CRITIS 2006. LNCS, vol. 4347, pp. 235–248. Springer, Heidelberg (2006)
5. Cox Jr, L.A.: Game theory and risk analysis. Risk Anal. **29**(8), 1062–1068 (2009)
6. Cremonini, M., Martini, P.: Evaluating information security investments from attackers perspective: the return-on-attack (ROA). In: 4th Workshop on the Economics on Information Security (2005)
7. Kordy, B., Kordy, P., Mauw, S., Schweitzer, P.: ADTool: security analysis with attack–defense trees. In: Joshi, K., Siegle, M., Stoelinga, M., D'Argenio, P.R. (eds.) QEST 2013. LNCS, vol. 8054, pp. 173–176. Springer, Heidelberg (2013)
8. Kordy, B., Piètre-Cambacédès, L., Schweitzer, P.: DAG-based attack and defense modeling: don't miss the forest for the attack trees. Comput. Sci. Rev. **13–14**, 1–38 (2014)

9. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 186–198. Springer, Heidelberg (2006)
10. de la Maza, M., Tidor, B.: An analysis of selection procedures with particular attention paid to proportional and Boltzmann selection. In: Proceedings of the 5th International Conference on Genetic Algorithms, pp. 124–131 (1993)
11. Nulton, J.D., Salamon, P.: Statistical mechanics of combinatorial optimization. Phys. Rev. A **37**(4), 1351–1356 (1988)
12. Pieters, W.: Defining "the weakest link": comparative security in complex systems of systems. In: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), vol. 2, pp. 39–44, December 2013
13. Pieters, W., Lukszo, Z., Hadžiosmanović, D., Van den Berg, J.: Reconciling malicious and accidental risk in cyber security. J. Internet Serv. Inf. Secur. **4**(2), 4–26 (2014)
14. Pieters, W., Van der Ven, S.H.G., Probst, C.W.: A move in the security measurement stalemate: elo-style ratings to quantify vulnerability. In: Proceedings of the 2012 New Security Paradigms Workshop, NSPW 2012, pp. 1–14. ACM (2012)
15. Schneier, B.: Attack trees: modeling security threats. Dr. Dobb's J. **24**(12), 21–29 (1999)
16. The Open Group. Risk taxonomy. Technical report C081, The Open Group (2009)