Processing Multi-Parameter Attacktrees with Estimated Parameter Values

Aivo Jürgenson[1,2] and Jan Willemson[3,4]

[1] Tallinn University of Technology, Raja 15, 12618 Tallinn, Estonia.
aivo.jurgenson@eesti.ee
[2] Elion Enterprises Ltd, Endla 16, 15033 Tallinn, Estonia.
[3] Tartu University, Institute of Computer Science, Liivi 2, Tartu, Estonia. jan@ut.ee
[4] Cybernetica, Aleksandri 8a, Tartu, Estonia

Abstract. Authors extend the multi-parameter attacktree model to include inaccurate or estimated parameter values, which are modelled as probabilistic interval estimations. The paper develops mathematical tools to extend the computation rules of the attacktree model to work with interval estimations instead of point estimates. We present a sample computation routine and discuss how to interpret the analysis results and how to choose the optimal or an economically justified security level.

1 Introduction

Recent developments in information technology have changed the way we live and work. We can communicate faster and in larger volumes than ever before, our productivity has increased dramatically due to task automation and parallelization, etc. Unfortunately, information technology has also helped the dark side. Besides legitimate work, attacking someone's (digital) assets has become much more efficient as well. Properties of the digital world make attacks highly parallelizable, the traces easily hidable and the damage occurring almost instantly.

Thus, in order to perform one's duties under such conditions, applying adequate security mechanisms becomes a necessary prerequisite. Still, the number of possible attack countermeasures is large and the task of picking the right set is far from being trivial. Accordingly, there are several approaches for this task.

Parker [1] emphasizes the importance of planning and selecting information safeguards to achieve due diligence toward achieving enablement of trustworthy business and engineering systems and compliance with the regulations and legislation. For Parker, compliance is nowadays

more important than security risk reduction considering the penalties being applied to organizations that fail to meet regulatory requirements.

The authors of the current paper however feel that even though compliance to regulatory requirements may ensure a *sufficient* security level, several aspects of security management remain uncovered. For example, business management usually does not only require security level to be sufficient, but also *optimal* in the sense that no over exaggerated investments have been made. Thus, a good security manager must be able to explain to the board, what the company is getting in return for the money invested into security [2,3].

This question is very hard to answer without a thorough risk assessment. Thus, developing methods for IT risk analysis is a major challenge requiring a solution when building large computer-dependent infrastructures.

Since attacks are human created and constantly evolving, it is not possible to establish any fully automatic risk analysis mechanisms (even though several promising approaches exist based on attack graphs [4,5]). Expert knowledge will always play a substantial role with security analysis. However, expert evaluations are generally rather rough and can not cope with very complicated threats. For instance, [6] provides us with approximate expert-assigned probabilities of a number of threats with the precision of 0.1 on the scale $0 \ldots 1$. Even with such level of roughness, we only see estimates for relatively simple events "Attempted Unauthorized System Access by Outsider", "Abuse of Access Privileges by Other Authorized User", etc., but not for complicated scenarios like "Loss of profits due to lost user base after online service inaccessibility for 5 hours".

Thus a method needs to be developed that on one hand is able to handle complicated threats, but at the same time could still make use of approximate expert knowledge. In this paper we study a method related to attack graphs as considered by Sheyner et. al. [5,7,8], where one is interested in describing and comparing different event sequences that will eventually result in successful penetration of security mechanisms. However, we will assume some extra structure from these graphs so that different subattacks are organized hierarchically and form a tree. The next Section will cover this issue in more detail.

2 Hierarchical Security Assessment – State of the Art

In order to better assess security level of a complex and heterogeneous system, a gradual refinement method called *threat tree* or *attack tree method* can be used. Basic idea of the approach is simple – the analysis begins by identifying one or more *primary threats* and continues by splitting the threat into subattacks, either all or some of them being necessary to materialize the primary threat. The subattacks can be split further etc., until we reach the state where it no more makes sense to split the resulting attacks any further; these kinds of non-splittable attacks are called *elementary* or *atomic attacks* and the security analyst will have to evaluate them somehow. During the splitting process, a tree is formed having the primary threat in its root and elementary attacks in its leaves. Using the structure of the tree and the estimations of the leaves, it is then (hopefully) possible to give some estimations of the root node as well.

Threat tree approach to security evaluation is several decades old. It has been used for tasks like fault assessment of critical systems [9] or software vulnerability analysis [10,11], and was adapted to information security by Bruce Schneier [12,13].

Earlier works in this field considered attack trees using only one estimated parameter like cost or feasibility of the attack, skill level required, etc. [11,12,14]. Opel [15] considered also multi-parameter attack trees, but the actual tree computations in his model still used only one parameter at a time. Even though single-parameter attack trees can capture some aspects of threats reasonably well, they still lack the ability to describe the full complexity of the attacker's decision-making process.

A substantial step towards better understanding the motivation of the attacker was made in 2006 by Buldas et. al. [16]. Besides considering just the cost of the attack, they also used success probability together with probabilities and amount of penalties in the case of success or failure of the attack in their analysis. As a result, a much more accurate model of the attack game was obtained.

The model of [16] has a significant drawback when it comes to practical application. Namely, the authors of [16] consider all the parameter values to be precise point estimates. Still, in practice security analysts rarely tend to provide exact numerical values for costs, probabilities etc. Instead, it feels much more natural to talk about intervals where the parameters belong to with some confidence.

The purpose of this paper is to extend the research of Buldas et. al. [16] by replacing exact values with interval estimations. The paper is organized as follows. First, in Section 3 we give a more formal definition of attack trees. In order to be able to give estimations of higher level attacks based on more elementary ones, rules of computation with interval estimations must be developed and this is done in Section 4. Section 5 presents an example of the computation routine and gives general rules for result interpretation. Finally, in Section 6 we draw some conclusions and give directions for future work.

3 Attack Trees

As mentioned in Section 2, attack tree is a result of a top-down process of splitting complex attacks into simpler ones. In this paper, we will consider two types of splits – AND-splits and OR-splits.[5] Thus, there are altogether three types of nodes in the tree.

1. *Leaf node* or *elementary attack*, which does not have any subattacks and which success does not depend on any other attacks. The parameter values of the leaf nodes are assigned by the experts.
2. *OR-node*, which has child nodes; for the OR-node attack to succeed, at least one of the sub-attacks must succeed.
3. *AND-node*, which has child nodes; for the AND-node attack to succeed, all of the sub-attacks must succeed as well.

Following [16], we will use the parameters in Table 1 that are to be evaluated in the leaf nodes and computed throughout the tree.

It will later prove useful to denote the expected loss in case the attack was successful $q_S \cdot k_S$ by π_S and the expected loss in case the attack was not successful $q_F \cdot k_F$ by π_F.

We will denote the cost of the elementary attack A as $\mathsf{Cost}(A)$ and similar notation will be used for other parameters as well.

We will later use the example attack tree given in Figure 1 describing a simple security analysis of information leak from a company. The tree has four leaf nodes, two OR-nodes (one of them denoting the primary threat at the root) and one AND-node. We are using notion similar

[5] Even though the approach using only AND and OR splits is not flexible enough to cover all possible security settings (e.g. threshold security), they have proven to be enough in all practical threat trees analyzed by the authors.

Table 1. Parameters of the attacks

Parameter	Definition
Cost	cost of performing the attack
p	probability of success of the attack
q_S	probability of catching the attacker, if the attack was successful
k_S	penalty of the attacker, if the attack was successful and attacker was caught
q_F	probability of catching the attacker, if the attack was not successful
k_F	penalty of the attacker, if the attack was not successful and attacker was caught

to AND-gates ($\boxed{\&}$) and OR-gates ($\boxed{\geq 1}$) to distinguish between AND-nodes and OR-nodes.

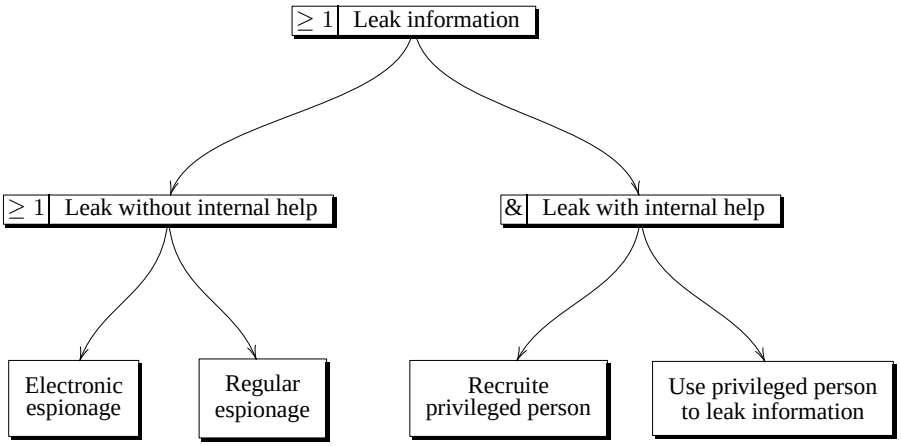

Fig. 1. Attacktree of leaking sensitive information from a company.

3.1 Tree Computations

The authors of [16] give the following formulae for computing the parameters of parent node C based on the values of child nodes A and B. If C is an AND-node, we get

$$\mathsf{Cost}(C) = \mathsf{Cost}(A) + \mathsf{Cost}(B), \tag{1}$$

$$p(C) = p(A) \cdot p(B), \tag{2}$$

$$\pi_S(C) = \pi_S(A) + \pi_S(B), \tag{3}$$

$$\begin{aligned}\pi_F(C) = \tfrac{1}{1-p(A)p(B)} \cdot [&p(A)(1-p(B))(\pi_S(A)+\pi_F(B)) + \\ &+(1-p(A))p(B)(\pi_F(A)+\pi_S(B)) + \\ &+(1-p(A))(1-p(B))(\pi_F(A)+\pi_F(B))].\end{aligned} \tag{4}$$

The following formula is used in case the node C is an OR-node.

$$\begin{aligned}&(\mathsf{Cost}(C), p(C), \pi_S(C), \pi_F(C)) \\ &= \begin{cases} (\mathsf{Cost}(A), p(A), \pi_S(A), \pi_F(A)) & \text{if } \mathsf{Outcome}(A) \geq \mathsf{Outcome}(B) \\ (\mathsf{Cost}(B), p(B), \pi_S(B), \pi_F(B)) & \text{if } \mathsf{Outcome}(A) < \mathsf{Outcome}(B) \end{cases},\end{aligned} \tag{5}$$

where $\mathsf{Outcome}(A)$ is the outcome of the attack A for attacker. Its value is computed as

$$\mathsf{Outcome}(A) = p(A)\cdot\mathsf{Gain} - p(A)\cdot\pi_S(A) - (1-p(A))\cdot\pi_F(A) - \mathsf{Cost}(A)\ , \tag{6}$$

where Gain is a global parameter expressing the total gain of the attacker in case the primary threat is materialized.

4 Modeling Parameter Estimations

As discussed above, security experts may find it more comfortable working with intervals, rather than exact values. When talking about the value of some parameter belonging to an interval, such claims are usually not absolute, but hold with some level of confidence. So we can write

$$p_X = \Pr[k_1 \leq X \leq k_2]\ , \tag{7}$$

where p_X is the *probability* of the unknown value of the parameter X being within the interval of $[k_1, k_2]$. We will later refer to p_X as *confidence* or *confidence level* and $\mathcal{X} = (p_X, k_1, k_2)$ as *estimation*. The set of all estimations will be denoted as $\mathbb{P}$.

4.1 Motivation and Connections with Bayesian Networks

In order to handle the estimations in intuitively comprehensible manner, we will consider estimated parameters as random variables. The probabilistic inference between random variables has been extensively studied in the Bayesian Networks (BN) theory and used e.g. in the fields of artificial intelligence and machine learning. Attack trees can be viewed as a special kind of BN graphs, which try to investigate the likelihood of the primary threats, given the information about leaf attacks. Attack tree structure represents causality connections between attacks and the node parameters represent random variables in the BN graph. One way to ”convert” attack trees to the causal networks has been presented by Qin and Lee [17]. In case of multi-parameter attack trees, each node has many variables and the inference between nodes is a bit more complex, as expressed by the formulae (1) – (6).

The general structure of operations on estimations follows a simple pattern – given argument estimations, we first convert them to random variables, then perform our computation operations and then convert the resulting random variable back to the estimation based on its distribution. In order to do the first conversion, we need some assumptions about the corresponding distributions, and in the current paper we will take all our distributions to be normal. Additionally, we assume that all our estimations in leaf nodes and therefore the random variables as well are independent of each other, which in practice is roughly usually the case. This allows us to simplify the operations on random variables.

The assumption about normal distributions is natural for most of the parameters, since security experts evaluating them are humans and humans tend to estimate values using normal distribution. The concept of using imprecise data and estimations in the BN graphs is not new and has been explored in [18] and [19]. However, in this paper, we do not try to compute the exact (conditional) distributions of all our variables, but simply use normal distributions as is generally done when trying to simplify the BN calculations. Note that we merely use the normality assumption as a heuristic that helps our expert to deduce estimations of parent nodes based on estimations of the child nodes in the attack tree, and we make no formal claims concerning what these distributions in reality might be.

One can think of the analyst’s task to find out whether the system is vulnerable with respect to the primary threat. While completing this task,

the analyst considers the leaves of his current tree (which just consists of one root node in the beginning of the process) and if (s)he is unable to evaluate some parameters of some leaf, (s)he must develop it further. After "solving" the resulting subtree, only the estimations of the parameters of the current node are important, and several heuristics can be used to achieve a reasonable result. The heuristic assuming normal distributions is just one possibility; other possibilities definitely exist, but they remain outside of the scope of the current paper.

There is another detail to note. Some of our parameters have fixed value domains, e.g. are probabilities and hence belong to $[0, 1]$, so we can not claim that they follow normal distribution. However, considering probabilities itself as imprecise values (second-order probability distribution) is well known in Bayesian statistics and for our estimations it is enough that $[k_1, k_2] \subset [0, 1]$. There is no harm caused when the corresponding parameter is internally interpreted as a normally distributed random variable, even if its original value represents a probability. We will cover this issue in more detail in the end of Section 4.

4.2 Estimation Arithmetic

Our goal is to replace exact parameter values in formulae (1) – (6) by estimations. To do so, we will have to define addition, subtraction, multiplication, division and comparison of estimations, but also multiplication by and adding to a real number.

To use estimations in our formulae, we next discuss how to define the required operations in such a way that $\mathbb{P}$ would become closed under these operations.

Conversion between estimations and random variables To convert the estimation $\mathcal{X}$ to a random variable X, we have to find out the mean a_X and standard deviation σ_X. From the assumption above and from equation (7) we can get the following formulae:

$$a_X = \mathbf{E}X = \frac{k_1 + k_2}{2} \ , \tag{8}$$

$$p_X = \Pr(k_1 \leq X \leq k_2) = \Phi\left(\frac{k_2 - a_X}{\sigma_X}\right) - \Phi\left(\frac{k_1 - a_X}{\sigma_X}\right) \ , \tag{9}$$

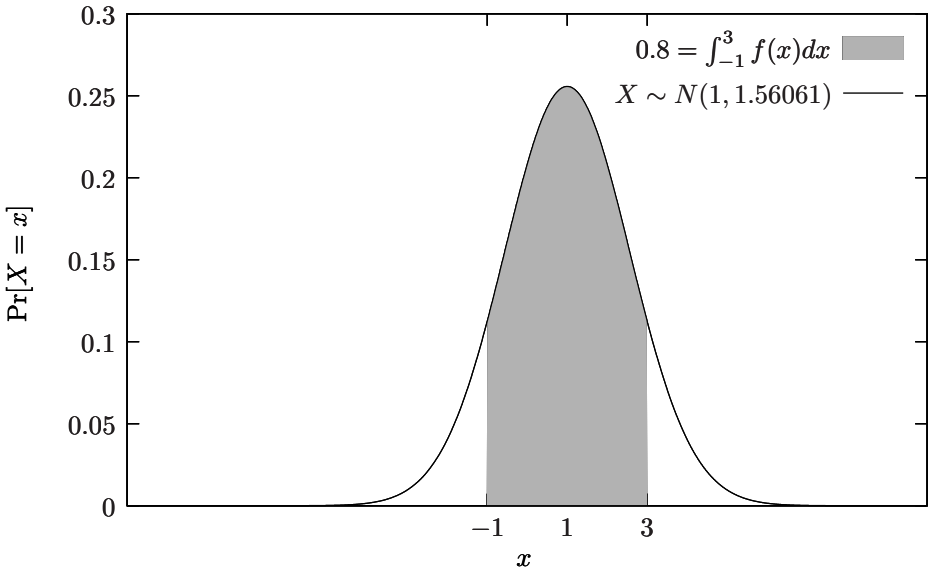

Fig. 2. Conversion $\mathcal{X} = (0.8, -1, 3) \rightarrow X \sim N(1, 1.56061)$.

where the $\Phi(x)$ is the Laplace's function. Although we cannot give explicit formula for calculating σ_X, we can certainly solve the equation (9) to compute the standard deviation σ_X using a computer.

We denote conversion of estimation $\mathcal{X}$ to normally distributed random variable X as $\mathcal{X} = (p_X, k_1, k_2) \rightarrow X \sim N(a_X, \sigma_X)$. An example is depicted in Figure 2 for the conversion $\mathcal{X} = (0.8, -1, 3) \rightarrow X \sim N(1, 1.56061)$.

To convert the probabilistic variable X back to an estimation $\mathcal{X}$, we would need to know the confidence p'_X, at which we would like to express the estimation. After we have specified p'_X, we can compute the appropriate interval $[k_1, k_2]$ from equation (9). We denote such back-conversion as $X \sim N(a_X, \sigma_X) \rightarrow \mathcal{X} = (p'_X, k'_1, k'_2$.)

To simplify the operations with our estimations of the attack-tree node parameters, we will convert all estimations to the same confidence level p_T, which will be defined globally for the attack-tree. In effect, p_T defines the confidence level or the margin of error at which we would like to have the answer of our attack-tree analysis given. If the original estimation $\mathcal{X}$ of an expert is given using some other confidence level p_X,

we first convert $\mathcal{X} = (p_X, k_1, k_2) \to X \sim N(a_X, \sigma_X)$ and then find the new interval $[k_1', k_2']$ by $X \sim N(a_X, \sigma_X) \to \mathcal{X} = (p_T, k_1', k_2'$.)

Next, we need to define the computation rules for estimations, i.e. operations $+$ and $\cdot$ as functions of type $\mathbb{IP} \times \mathbb{IR} \to \mathbb{IP}$, operations $+$, $\cdot$ and $/$ as functions of type $\mathbb{IP} \times \mathbb{IP} \to \mathbb{IP}$ and a binary relation $\geqq$ on the set $\mathbb{IP}$.

Adding a real number to an estimation Given $r \in \mathbb{IR}$ together with estimation $\mathcal{X} = (p_T, k_1, k_2)$ and wanting to compute $\mathcal{X} + r = \mathcal{Y}$, we first convert $\mathcal{X} \to X \sim N(a_X, \sigma_X)$. After that we can compute $Y = r + X$ using the properties $a_Y = r + a_X$ and $\sigma_Y = \sigma_X$. Finally, we have $Y \to \mathcal{Y} = (p_T, k_1 + r, k_2 + r)$.

Multiplying an estimation by a real number It is known that given $X \sim N(a_X, \sigma_X)$ and $r \in \mathbb{IR}$ we have $r \cdot X = Y \sim N(r \cdot a_X, |r \cdot \sigma_X|)$. To simplify the computation, $\mathcal{X} = (p_T, k_1, k_2)$ is first centralized to $\mathring{\mathcal{X}} = \mathcal{X} - a_X$. Then $\sigma_Y = |r \cdot \sigma_X|$ and now we obtain $\mathring{Y} \to \mathring{\mathcal{Y}} = (p_T, -k, k)$, where the interval $[-k; k]$ is found from the equation

$$p_T = 2\Phi\left(\frac{|k|}{\sigma_X}\right) \quad . \tag{10}$$

After that $\mathring{\mathcal{Y}}$ is de-centralized by shifting its interval by $r \cdot a_X$. Thus, we finally get $\mathcal{Y} = (p_T, -k + r \cdot a_X, k + r \cdot a_X)$.

Adding two estimations $\mathcal{X}_1$ and $\mathcal{X}_2$ When adding two estimations $\mathcal{X}_1 = (p_T, k_1, k_2)$ and $\mathcal{X}_2 = (p_T, k_1', k_2')$, we first convert both of them to $X_1 \sim N(a_1, \sigma_1)$ and $X_2 \sim N(a_2, \sigma_2)$. Then, assuming X_1 and X_2 to be independent, we can compute $Y = X_1 + X_2$, where $a_Y = a_1 + a_2$ and $\sigma_Y = \sigma_1 + \sigma_2$. In case of subtracting, we get $a_Y = a_1 - a_2$, but $\sigma_Y = \sigma_1 + \sigma_2$.

Now we have the necessary information to convert $Y \to \mathcal{Y}$. First we convert $\mathring{Y} \to \mathring{\mathcal{Y}} = (p_T, -k, k)$, where k is found from equation (10) by replacing σ_X with σ_Y. Now, $\mathring{\mathcal{Y}}$ can be de-centralized by shifting its interval by $a_1 + a_2$. So, we get the final result $\mathcal{Y} = (p_T, -k + a_1 + a_2, k + a_1 + a_2)$, or in case of the subtraction, $\mathcal{Y} = (p_T, -k + a_1 - a_2, k + a_1 - a_2)$.

Multiplying two estimations $\mathcal{X}_1$ and $\mathcal{X}_2$ When multiplying two estimations $\mathcal{X}_1$ and $\mathcal{X}_2$ we first centralize them to $\mathring{\mathcal{X}}_1$ and $\mathring{\mathcal{X}}_2$. Then $\mathring{Y} =$

$\mathring{X}_1 \cdot \mathring{X}_2$ is calculated using the assumption of independent variables and the fact that $\sigma_Y = \sigma_1 \cdot \sigma_2$, however, the distribution of $\mathring{Y}$ is not normal, but Bessel distribution.

To convert Y to $\mathcal{Y}$, we compute $\mathring{Y} \rightarrow \mathring{\mathcal{Y}} = (p_T, -k_B, k_B)$, where k_B is found from equation

$$p_T = \Pr[-k_B \leq Y \leq k_B] = \int_{-k_B}^{k_B} f_B(y, \sigma_Y) dy \ , \tag{11}$$

where

$$f_B(y, \sigma_Y) = \frac{1}{\pi \sigma_Y} K_0 \left(\frac{|y|}{\sigma_Y} \right)$$

is the probability density function of the Bessel distribution. Now, $\mathring{\mathcal{Y}}$ needs to be de-centralized. The mean of $\mathcal{Y}$ could be computed as $a_Y = \mathbf{E}(X_1 \cdot X_2) = \mathbf{E}X_1 \cdot \mathbf{E}X_2 = a_1 \cdot a_2$. Therefore, we can shift $\mathring{\mathcal{Y}}$ interval by $a_1 \cdot a_2$. So, we get the final result $\mathcal{Y} = (p_T, -k_B + a_1 \cdot a_2, k_B + a_1 \cdot a_2)$.

Dividing two estimations $\mathcal{X}_1$ and $\mathcal{X}_2$ Using centralized independent variables $\mathring{X}_1$ and $\mathring{X}_2$, it is known that $\mathring{X}_1 / \mathring{X}_2 = \mathring{Y} \sim \text{Cauchy}(0, \sigma_Y)$, where $\sigma_Y = \frac{\sigma_1}{\sigma_2}$.

Using the global confidence value p_T, we convert the Y to $\mathcal{Y}$. In order to do that, we first convert $\mathring{Y} \rightarrow \mathring{\mathcal{Y}} = (p_T, -k_C, k_C)$, where k_C is found from the equation

$$p_T = \Pr[-k_B \leq Y \leq k_B] = \int_{-k_B}^{k_B} f_C(y, y_0, \sigma_Y) dy \ , \tag{12}$$

where

$$f_C(y, y_0, \sigma_Y) = \frac{\sigma_Y}{\pi} \cdot \frac{1}{\sigma_Y^2 + (y - y_0)^2}$$

is the probability density function of the Cauchy distribution. Now, $\mathring{\mathcal{Y}}$ needs to be de-centralized. The mean of $\mathcal{Y}$ can be computed as $a_Y = \mathbf{E}(X_1 / X_2) = \mathbf{E}X_1 / \mathbf{E}X_2 = a_1 / a_2$. Therefore, we can shift $\mathring{\mathcal{Y}}$ interval by a_1 / a_2. So, we get the final result $\mathcal{Y} = (p_T, -k_C + a_1/a_2, k_C + a_1/a_2)$.

Comparing two estimations $\mathcal{X}_1$ and $\mathcal{X}_2$ The last operation to enable us to use estimations as operands in our formulae is the comparison. To decide the order of the estimations, we compare the means of the

corresponding random variables. Formally, we can define the comparison as the binary relation $\geqq = \{(\mathcal{X}_1; \mathcal{X}_2) | \mathcal{X}_1 = (p_T, k_1, k_2) \rightarrow X_1 \sim N(a_1, \sigma_1), \mathcal{X}_2 = (p_T, k_1', k_2') \rightarrow X_2 \sim N(a_2, \sigma_2), a_1 \geq a_2\}$.

4.3 Soundness of computations

Almost all parameters of the nodes have a limited value domain predefined by the interpretation of the parameter, e.g. Cost should be a non-negative real number and p is a probability belonging to the interval $[0, 1]$. When estimations are considered instead of specific values, it is still natural to assume that the respective intervals $[k_1, k_2]$ are subsets of the value domains (e.g. $[k_1, k_2] \subset [0, \infty)$ for Cost and $[k_1, k_2] \subset [0, 1]$ for p). Even if expert estimations given to leaf node parameters satisfy these assumptions, it may happen that as a result of tree computations, some of the parameters in other nodes do not.

Generally, such a situation indicates that no sound conclusions can be drawn on the given confidence level p_T. This problem can be solved in a number of ways.

- The global confidence level p_T can be decreased in order to achieve soundness of estimations in all the nodes. It is possible to find the largest value p_T ensuring sound conclusions and this value can be considered as the confidence level of the whole tree.
- It is possible to define the required confidence level locally for each node.
- It is possible to adjust one or both of the bounds k_1 and k_2 to fit into the required interval; this will automatically decrease the confidence level of the respective node as well.

Each of these approaches has its pros and cons; selecting the best one may be application specific and remains the subject for future research.

5 Tree Computations with Estimations

First consider as an illustration a simple attack tree computation routine based on the example given in Figure 1. First we fix the level of confidence of our estimations to be $p_T = 0.9$ and second we let out experts to evaluate the parameters of the leaves with this confidence. Assume we get the estimation interval for the Cost parameter in the leaf “Use privileged

person to leak information" at this confidence to be $[7.56 \cdot 10^4, 1.24 \cdot 10^5]$, etc, as given in Table 2.

Next we use the computation rules developed in Section 4 to obtain parameter estimations for non-leaf nodes as well (see Table 2). Finally, in the root node we use (6) in its interval form to find $\Pr[2.51 \cdot 10^7 \leq \mathsf{Outcome} \leq 5.36 \cdot 10^7] = 0.9$, which shows that the outcome of the attack is with high probability positive for the attacker, hence some measures must be introduced in order to counter it.

Table 2: Attacktree of leaking sensitive information from a company.

ID	**Description**	**Type**	**Parameter estimations**
	Gain of the attacktree		$\Pr[5.23 \cdot 10^7 \leq \mathsf{Gain} \leq 2.48 \cdot 10^8] = 0.9$
A	Leak information	OR	$\Pr[8.58 \cdot 10^4 \leq \mathsf{Cost} \leq 2.14 \cdot 10^5] = 0.9$
			$\Pr[0.0561 \leq p \leq 0.544] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_S \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_F \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[2.51 \cdot 10^7 \leq \mathsf{Outcome} \leq 5.36 \cdot 10^7] = 0.9$
A.1	Leak without internal help	OR	$\Pr[8.58 \cdot 10^4 \leq \mathsf{Cost} \leq 2.14 \cdot 10^5] = 0.9$
			$\Pr[0.0561 \leq p \leq 0.544] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_S \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_F \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[2.51 \cdot 10^7 \leq \mathsf{Outcome} \leq 5.36 \cdot 10^7] = 0.9$
A.1.1	Electronic espionage	LEAF	$\Pr[9.86 \cdot 10^6 \leq \mathsf{Cost} \leq 1.11 \cdot 10^7] = 0.9$
			$\Pr[0.186 \leq p \leq 0.314] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_S \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_F \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[1.71 \cdot 10^7 \leq \mathsf{Outcome} \leq 2.59 \cdot 10^7] = 0.9$
A.1.2	Regular espionage	LEAF	$\Pr[8.58 \cdot 10^4 \leq \mathsf{Cost} \leq 2.14 \cdot 10^5] = 0.9$
			$\Pr[0.0561 \leq p \leq 0.544] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_S \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_F \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[2.51 \cdot 10^7 \leq \mathsf{Outcome} \leq 5.36 \cdot 10^7] = 0.9$
A.2	Leak with internal help	AND	$\Pr[8.66 \cdot 10^6 \leq \mathsf{Cost} \leq 2.15 \cdot 10^7] = 0.9$
			$\Pr[0.0779 \leq p \leq 0.0871] = 0.9$
			$\Pr[9.72 \cdot 10^6 \leq \pi_S \leq 1.23 \cdot 10^7] = 0.9$
			$\Pr[-3.33 \cdot 10^6 \leq \pi_F \leq 2.53 \cdot 10^7] = 0.9$
			$\Pr[-2.05 \cdot 10^7 \leq \mathsf{Outcome} \leq -6.98 \cdot 10^6] = 0.9$
A.2.1	Recruite privileged person	LEAF	$\Pr[8.58 \cdot 10^6 \leq \mathsf{Cost} \leq 2.14 \cdot 10^7] = 0.9$
			$\Pr[0.0858 \leq p \leq 0.214] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_S \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[4.86 \cdot 10^6 \leq \pi_F \leq 6.14 \cdot 10^6] = 0.9$
			$\Pr[-8.16 \cdot 10^6 \leq \mathsf{Outcome} \leq 1.22 \cdot 10^7] = 0.9$

continues next page ...

Table 2 — continues from previous page ...

ID	Description	Type	Parameters estimations
A.2.2	Use privileged person to leak information	LEAF	$\Pr[7.56 \cdot 10^4 \leq \mathsf{Cost} \leq 1.24 \cdot 10^5] = 0.9$ $\Pr[0.428 \leq p \leq 0.672] = 0.9$ $\Pr[4.86 \cdot 10^6 \leq \pi_S \leq 6.14 \cdot 10^6] = 0.9$ $\Pr[4.86 \cdot 10^6 \leq \pi_F \leq 6.14 \cdot 10^6] = 0.9$ $\Pr[6.98 \cdot 10^7 \leq \mathsf{Outcome} \leq 8.4 \cdot 10^7] = 0.9$

In general, we may have three possible classes of estimations $\mathcal{X} = (p_X, k_1, k_2)$ for Outcome of the root node:

1. $0 < k_1 < k_2$, in which case we say that the vulnerability level of the primary threat under consideration with respect to the required confidence level is *high*;
2. $k_1 < k_2 < 0$, in which case we say that the vulnerability level is *low*;
3. $k_1 \leq 0 \leq k_2$, in which case we say that the vulnerability level is *medium*.

If needed, the last class can be further divided into lower medium and higher medium vulnerability levels depending on whether the mean value $\frac{k_1+k_2}{2}$ of the estimation (considered as a normally distributed random variable) is less or greater than zero.

If the security analyst finds out that the security level is not acceptable, (s)he concludes that some measures must be implemented. The possible measures are usually targeted towards lowering attack success probability or increasing expected penalties (e.g. by increasing probability of getting caught). When some set of protection measures is considered, the tree computations can be performed again for a new setting and if the security level becomes acceptable, we know that the set of measures is sufficient. It only remains to pick the most adequate set of such measures (this step is identical to the one described in [16]).

Following the tree computation routine, we can also find out which nodes of the tree are critical ones and must be addressed with our security enhancements. E.g. in the example presented above we see that parameters of the root node are derived from the parameters of the leaf "Regular espionage", which is thus the most vulnerable node in this setting.

6 Conclusions and Further Work

We presented an extension of the multi-parameter threat tree model to the case where the parameters of elementary attacks are given as inter-

val estimations rather than exact values. Clearly, such a problem setting implies the need to compute the values of primary threat parameters as estimations as well. A suitable method for defining necessary algebraic operations and relations on evaluations was developed in this paper and illustrated with a simple attack scenario.

There are still several directions our research can be continued in. First, our current heuristic used to compute with estimations is to consider all the parameters as normally distributed with suitable parameters. This simplification can turn out to be too restrictive for some applications, hence further studies are needed to find out how other distributions behave under the given tree computation rules.

Second, our tree computation routine can give out-of-bounds values for some parameters (e.g. probabilities) in some nodes. There are several possible solutions to this problem and selecting the best one remains the subject for future research as well.

And, last but not least, even though the authors have used attack tree approach successfully in several security analyses, its extension to interval estimations still needs further practical evaluation.

7 Acknowledgments

This research has been supported by the Estonian Science Foundation grant no. 7081.

References

1. Parker, D.B.: Fighting Computer Crime: A New Framework for Protecting Information. John Wiley & Sons (2001)
2. Geer, D., Hoo, K.S., Jaquith, A.: Information security: Why the future belongs to the quants. IEEE Security and Privacy **1**(4) (2003) 24–32
3. Sonnenreich, W., Albanese, J., Stout, B.: Return On Security Investment (ROSI) – A practical quantitative model. Journal of Research and Practice in Information Technology **38**(1) (February 2006) 55–66
4. Rieke, R.: Modelling and analysing network security policies in a given vulnerability setting. In López, J., ed.: CRITIS '06. Volume 4347 of LNCS., Springer Verlag (2006) 67–78
5. Sheyner, O., Wing, J.: Tools for generating and analyzing attack graphs. In de Boer et al., F., ed.: Proceedings of Workshop on Formal Methods for Components and Objects. Volume 3188 of LNCS., Springer Verlag (2004) 344–371
6. Meritt, J.W.: A method for quantitative risk analysis. In: Proceedings of the 22nd National Information Systems Security Conference. (1999)

7. Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.: Automated generation and analysis of attack graphs. In: Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA (May 2002)
8. Jha, S., Sheyner, O., Wing, J.: Two formal analyses of attack graphs. In: Proceedings of the 15th IEEE Computer Security Foundations Workshop. (2002) 49–63
9. Vesely, W., Goldberg, F., Roberts, N., Haasl, D.: Fault Tree Handbook. US Government Printing Office (January 1981) Systems and Reliability Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission.
10. Viega, J., McGraw, G.: Building Secure Software: How to Avoid Security Problems the Right Way. Addison Wesley Professional (2001)
11. Moore, A.P., Ellison, R.J., Linger, R.C.: Attack modeling for information security and survivability. Technical Report CMU/SEI-2001-TN-001, Software Engineering Institute (2001)
12. Schneier, B.: Attack trees: Modeling security threats. Dr. Dobb's Journal **24**(12) (December 1999) 21–29
13. Schneier, B.: Secrets & Lies. Digital Security in a Networked World. John Wiley & Sons (2000)
14. Mauw, S., Oostdijk, M.: Foundations of attack trees. In Won, D., Kim, S., eds.: International Conference on Information Security and Cryptology – ICISC 2005. LNCS 3935, Springer (December 2005) 186–198
15. Opel, A.: Design and implementation of a support tool for attack trees. Technical report, Otto-von-Guericke University (March 2005) Internship Thesis.
16. Buldas, A., Laud, P., Priisalu, J., Saarepera, M., Willemson, J.: Rational Choice of Security Measures via Multi-Parameter Attack Trees. In: Critical Information Infrastructures Security. First International Workshop, CRITIS 2006. Volume 4347 of LNCS., Springer (2006) 235–248
17. Qin, X., Lee, W.: Attack plan recognition and prediction using causal networks. In: 20th Annual Computer Security Applications Conference. (December 2004) 370 – 379
18. Kleiter, G.D.: Propagating imprecise probabilities in bayesian networks. Artificial Intelligence **88**(1-2) (1996) 143–161
19. Borsotto, M., Zhang, W., Kapanci, E., Pfeffer, A., Crick, C.: A junction tree propagation algorithm for bayesian networks with second-order uncertainties. In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence. (2006) 455–464