Rational Choice of Security Measures via Multi-Parameter Attack Trees

Ahto Buldas[1,2,3,⋆], Peeter Laud[1,2], Jaan Priisalu[4], Märt Saarepera[5], and Jan Willemson[1,2,⋆⋆]

[1] Cybernetica, Akadeemia tee 21, Tallinn, Estonia
[2] University of Tartu, Liivi 2, Tartu, Estonia, {ahto.buldas,peeter_l,jan}@ut.ee
[3] Tallinn University of Technology, Raja 15, Tallinn, Estonia
[4] Hansapank, Liivalaia 8, Tallinn, Estonia, jaan.priisalu@hansa.ee
[5] Independent researcher, marts@neoteny.com

Abstract. We present a simple risk-analysis based method for studying the security of institutions against rational (gain-oriented) attacks. Our method uses a certain refined form of attack-trees that are used to estimate the cost and the success probability of attacks. We use elementary game theory to decide whether the system under protection is a realistic target for gain-oriented attackers. Attacks are considered unlikely if their cost is not worth their benefits for the attackers. We also show how to decide whether the investments into security are economically justified. We outline the new method and show how it can be used in practice by going through a realistic example.

1 Introduction

Rapid growth of society's dependence on computers and the Internet has drawn attention to the vulnerability of this technical infrastructure. Increasing numbers of IT security incidents all over the World have emphasized the importance of risk analysis methods capable of deciding whether an organization (e.g. a company) is sufficiently protected against attacks. The protection mechanisms are often costly, or at least, not for free. Managers of an organization would like the investments into security to be reasonable and worth their price. The security experts should, more and more often, explain to their managers what benefits exactly the organization is getting for the money that is invested into security [1, 2].

In contrast to the cryptographic techniques, the IT risk management techniques are still in an embryonic stage. This is one of the reasons of an increasing gap between theory and practice of information security [3]. Occasional stochastic risks (natural disasters, general criminal activity) can be evaluated rather easily, since there is enough statistical data concerning both the frequency (probability) and losses associated with such threats. Targeted gain-oriented attacks are much harder to model because their occurrence does not usually follow any reasonable statistical patterns and they tend to be rather victim-specific, which makes it difficult to find suitable risk metrics for attacks [1].

In the risk management field, risk is mostly defined as an expected loss, which is caused by *threats* – events that are considered bad (namely because they cause losses). Hence, the risk caused by threat $\mathcal{T}$ can be computed by $\mathsf{Risk}[\mathcal{T}] = \mathsf{Pr}[\mathcal{T}] \cdot \mathsf{Loss}[\mathcal{T}]$, where $\mathsf{Pr}[\mathcal{T}]$ denotes the probability of $\mathcal{T}$ and $\mathsf{Loss}[\mathcal{T}]$ denotes the associated loss. Hence, to estimate the security risk of a company we have to find all possible threats $\mathcal{T}$, to estimate the corresponding losses $\mathsf{Loss}[\mathcal{T}]$ and the probabilities $\mathsf{Pr}[\mathcal{T}]$, and finally to sum everything up

$$\mathsf{Risk} = \sum_{\mathcal{T}} \mathsf{Pr}[\mathcal{T}] \cdot \mathsf{Loss}[\mathcal{T}] \ . \tag{1}$$

⋆ Supported by Estonian Science Foundation grant #5870
⋆⋆ Supported by Estonian Science Foundation grant #6096

Once we are able to do so, the security management is a trivial task: (A) compute the risk by (1), (B) if the risk seems to be too high, introduce some measures and compute the risk again, (C) if the cost of the measures is lower than the difference of risks, then decide that the measures are worth their price. Otherwise, the measures are unreasonable because it would be more beneficial not to take any measures.

Unfortunately, such an approach is hard to adopt in practice. Even if we are able to estimate the losses associated with the threats, their probabilities are often very hard to judge. This is especially true for targeted attacks that for a given setting may occur only once. It is also the case that companies are rather reluctant to share information concerning their vulnerabilities and the previous security incidents. For some typical attacks there exist rough expert estimates [4]. However, such estimates can generally be given for elementary vulnerabilities, but not easily to the primary (loss causing) threats. For instance, in [4] we see estimates for the events "Attempted Unauthorized System Access by Outsider", "Abuse of Access Privileges by Other Authorized User", etc., but not for "Loss in Drop of Company's Shares due to Bad Publicity".

Thus, we need a methodology to deduce probabilities of complex attacks from the parameters of simple vulnerabilities. Note that it is insufficient to consider only the occurrence probabilities of the vulnerabilities, since the attacker may consider more parameters when deciding whether to attack or not (e.g. the probability of getting caught and the associated penalties).

One of the methods used in practical security analysis is the *threat tree* method, which has been used in several security-oriented tasks like fault assessment of critical systems [5] or software vulnerability analysis [6, 7], and was adapted to information security by Bruce Schneier [8, 9]. In order to apply this method, only the rational attackers are taken into account. As the latter ones attack only when the attack is profitable, their behavior can be modeled by estimating the cost of attacks. Threat trees help us when reasoning about the decision-making process of the attackers and they work by splitting complex attacks into simpler and easier to analyze sub-attacks. Hence they are suitable for computing costs and success probabilities of attacks and are useful tools for practical security management.

Even though the threat trees (also called attack trees to emphasize the attack modeling domain) can provide valuable insight to the system's security, their applications have been rather simplistic so far. Most of the reported studies only consider one specific parameter for the nodes like cost or feasibility of the attack, skill level required, etc. [8, 7, 10]. Opel [11] considers also multi-parameter attack trees, but the actual tree computations in his model still use only one parameter at a time.

However, it is the belief of the authors of the current paper that the actual decision-making process of attackers is more complicated and that the interactions between different parameters play an important role. For example, if the success probability and possible monetary gain are considered as parameters, their product (i.e. the expected gain) also has a meaning and can be taken into account when making decisions about attacks.

The main contribution of this paper is to study how threat trees behave and how tree computations must be done when several interdependent attack parameters are considered. In this paper, we will concentrate on the attacker's gain, the probability of success, the probability of getting caught and the possible penalties as the parameters, but the method we will develop is able to handle a much larger variety of multi-parameter sets. As an application of the attack tree computations, we will also demonstrate how to make rational decisions concerning the security measures.

The paper is organized as follows. In Section 2, we discuss the rational attackers paradigm and define attack trees. Section 3 states the main principles of attack analysis. Section 4 presents the threat tree method built on this analysis. In Section 5, we discuss the evaluation of security

measures and draw some conclusions in Section 6. Throughout the paper an illustrative example of a software-producing Company is discussed which helps the reader to get familiar with the terms and the methods stated.

2 Rational Attackers Paradigm and Attack Trees

Starting from this section we will assume the role of an attacker and try to model his decision-making process. Since in this paper we are interested in gain-oriented attacks, we will assume that *attackers behave in a rational way*. In particular, we assume that rational attackers

(1) do not attack if the attack-game is *unprofitable* and
(2) choose *the most profitable* ways of attacking, i.e. those with the highest outcome (see subsection 3.1).

This assumption is called *rational attacker's paradigm*. Based on this paradigm we can model the attacker's decision-making process. First, he needs to get an overview of all his measures (bribing victim's employees, gaining physical access, gaining network access, etc.). Second, he will combine his measures to come up with possible plans of attack, and third, he will evaluate all possible plans to find whether any of them is profitable, and if so, which one maximizes the profit. We will use attack trees to clarify such a process.

An attack tree is a compact graphical representation of all possible attack-plans. It is the outcome of a *gradual refinement procedure* that gives more and more detailed descriptions of the attacks until the *atomic attacks* are reached, the parameters (e.g. the cost) of which can be estimated without further refinements.

Each node of the attack tree represents an attack or a certain (probabilistic) condition whereas the root node represents the primary threat that we will try to analyze. The non-leaf nodes of the graph are AND nodes and OR nodes:

- The child nodes of an OR node represent a list of conditions (sub-attacks) each of which is sufficient for the attack (or threat) being successful.
- The child nodes of AND node represent a list of conditions (sub-attacks) each of which is necessary for the attack being successful. The leaves of the tree represent atomic attacks.

Definition 1. *A successful attack is a subtree T' such that: (1) T' contains the root node; (2) for any AND-node $v \in T'$, all child nodes of v belong to T'; (3) for any OR-node $v \in T'$, at least one child node of v belongs to T'.*

In order to illustrate the process of building an attack tree and also the future concepts presented later on, we will use an example of a software-producing Company that tries to protect its intellectual property from the competitors. The main setting is stated in Example 1.[6] The corresponding attack tree is depicted in Figure 1.

3 Main Principles of the Attack Analysis Method

This far, the threat-tree methods have mostly been used to determine the success probability and the cost of attacks [8, 7, 10]. These parameters are indeed important for the risk analysis but certainly not sufficient. The decision ("to attack" vs "not to attack") made by an attacker depends also

[6] The example covered throughout the paper is a simplified version of a real analysis performed by the authors of the paper in a real company. Due to confidentiality agreements, the identity of the company will not be presented here and all the numeric data is changed.

A software-producing Company considers as the main threat the situation where a competitor steals the code of the Company during the developing phase, completes it to a product and gets ”first to the market” advance. The result is a lost market share, which may cost a great deal. We call this threat a *forestalling release*.
There are two events necessary for a forestalling release: (A) The code is stolen by a competitor, **and** (B) The code is used in competitive products. In a simplified model, we consider three ways how the code can be stolen (Figure 1): (A1) via bribing a programmer, (A2) via network attack, **or** (A3) via physical (ordinary) robbery.
For a successful bribery attack (Figure 1), the attacker should: (A1.1) successfully bribe a programmer of the Company, **and** (A1.2) the bribed programmer should obtain the valuable code from the Company.
For a successful bribery attack (Figure 1), the attacker should: (A2.1) employ a hacker, (A2.2) the hacker should exploit a bug in the computer system, **and** (A2.3) there must be an exploitable bug in the computer system.
For a successful physical robbery (Figure 1), the attacker should: (A3.1) employ a robber, **and** (A3.2) the robber should successfully break into the Company and obtain the code.
A rational adversary should determine which of the three attacks is the most profitable and then perform this attack.

Example 1: The main example.

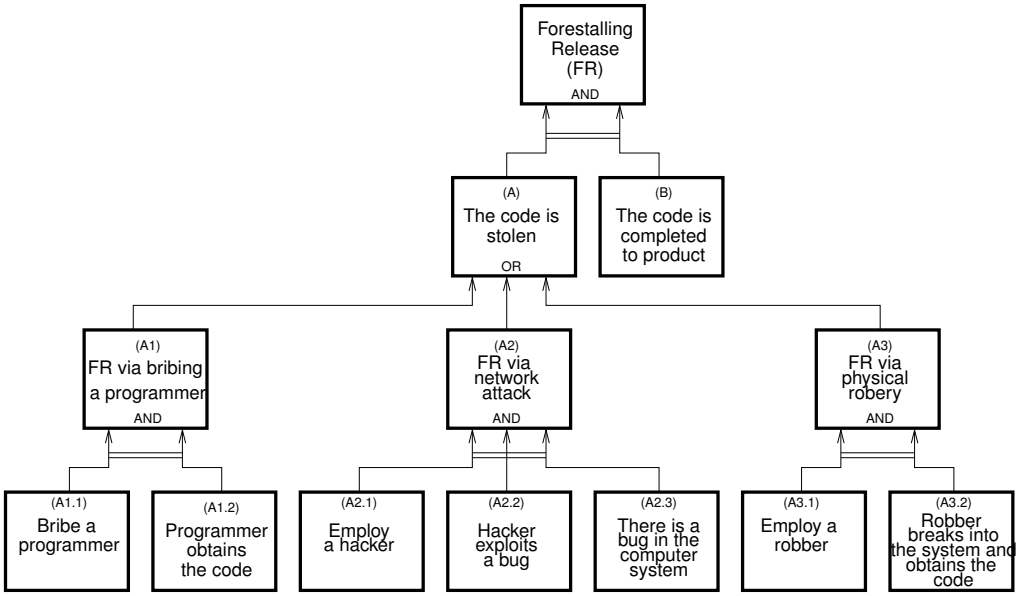

Fig. 1. A sample attack tree for a software developing company.

on the *attacker's risks*, i.e. on the probability that the attack will lead to a prosecution or penalty, as well as on the monetary losses that correspond to the prosecution or penalty.

Before describing our attack analysis method in detail (in Section 4), we outline the main starting points and principles, which the method is based on. In Subsection 3.1, we introduce a game-theoretic paradigm — the basis of the decision-making mechanism that we assume is used by rational attackers.

3.1 Attack as a Game

We view attack as a game played by the attacker. Rational attackers decide to play the attack-game if this is profitable for them. In order to decide about the profitability, the following parameters of the game will be taken into account:

- Gains – gains of the attacker, in case the attack succeeds
- Costs – cost of the attack

– p – success probability of the attack
– q – probability of getting caught (in case the attack was successful)
– Penalties – expected penalties in case the attacker is caught (assuming that the attack was successful)
– q_- – probability of getting caught (in case the attack was not successful)
– Penalties$_-$ – expected penalties in case the attacker is caught (assuming that the attack was not successful)

In our model, each attack begins with a *preparation* phase during which the attacker prepares the necessary resources for performing the attack (e.g. bribes some internal people from the Company, buys some attack time from a bot-net, etc.). After that, the attacker tries to break into the system. With probability p the attack is successful and the attacker obtains the Gains. In real life, it is possible that if later caught, the attacker may not be able to fully exploit the expected gains. However, for the sake of model simplicity we do not consider such a case here and will leave it for future research.

After the attack, it is possible (with probability q) that the attacker will be detected and get caught. We assume that in this case, the attacker has to pay Penalties.[7] The attacker may also get caught if the attack was unsuccessful, however, both the probability q_- of getting caught and penalties he has to pay (denoted by Penalties$_-$) are not necessarily equal to q and Penalties, respectively. [8]

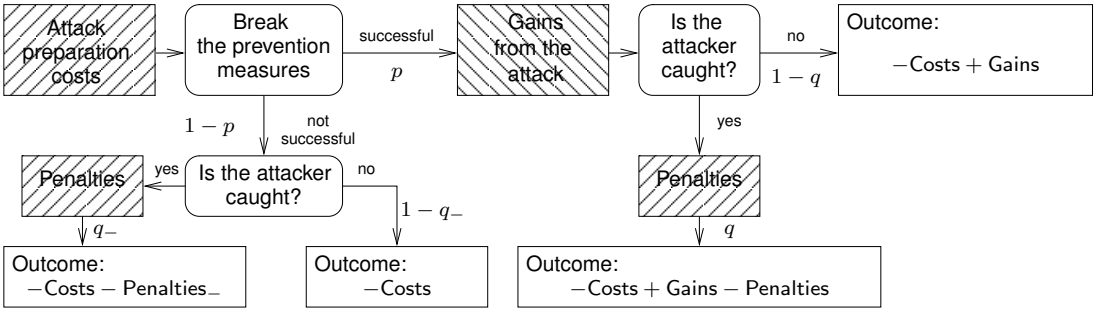

Fig. 2. Diagram (event tree) of the "attack game" from the attacker's point of view.

Figure 2 presents our model of attack in the form of an event-tree. The oval boxes represent (probabilistic) conditions or events, the dashed boxes denote the gains and losses of the adversary. The arrows represent the change of the state during the attack and they are labeled with the probabilities that the particular branch is chosen during the attack. The leaves of the tree represent the final outcome of the attacker. For example, if the attack is successful (with probability

[7] In practice it may happen that the probability of some penalties being enforced is rather low. In this case, we consider here the expected mean value of Penalties and Penalties$_-$.

[8] Note that the parameters Gains, Costs, Penalties and Penalties$_-$ are measured in monetary units. This approach differs from e.g. the one taken by Liu, Zang and Yu [12] who classify the risk evaluation criteria as cost and noncost constraints. However, the quantitaive nature of the methodology developed in the current paper presumes comparability of different attacker motives, and since for rational attackers most of the incentives are expressed in monetary units already, it natural to try to fit other attack targets (causing fear, achieving recognition in hacker community, etc.) into that scale, too. The authors note that since threats and attacks are more and more becoming trade articles (see also Schechter [13]), every attack will eventually have its true price. There are attack scenarios not fitting well into our model, most notably terrorism. However, according to CERT report from 2004 [14], only 1% of therats against information systems can be linked to terrorist motives.

Let the gains of the attacker be $\mathsf{Gains} = \$150,000$, the success probability of the attack be $p = 0.1$ and the cost be $\mathsf{Costs} = \$14,000$. If the attack is successful, then the probability of getting caught is $q = 0.01$ and the penalties are $\mathsf{Penalty} = \$200,000$. If the attack is not successful, then the corresponding probability and the penalty are $q_- = 0.1$ and $\mathsf{Penalty}_- = \$10,000$, respectively.
To decide, whether the attack is successful, we first compute the average penalties: $\pi = q \cdot \mathsf{Penalty} = 0.01 \cdot 200,000 = \$2000$, and $\pi_- = q_- \cdot \mathsf{Penalties}_- = 0.1 \cdot 10,000 = \$1000$. Second, we compute the expected outcome by using (2):

$$\mathsf{Outcome} = -\mathsf{Costs} + p \cdot (\mathsf{Gains} - \pi) - (1-p) \cdot \pi_- = -14,000 + 0.1(150,000 - 2000) - 0.9 \cdot 1000 = -\$100,$$

which means that the attack is not profitable to the attacker.

Example 2: Parameters of the attack.

p) and the attacker is caught (with probability q), then the final outcome of the attacker will be $-\mathsf{Costs} + \mathsf{Gains} - \mathsf{Penalties}$ and the probability of this particular branch is $p \cdot q$.

The overall *value of the game* (or expected outcome) for the attacker is

$$\begin{aligned}\mathsf{Outcome} = (1-p) \cdot [q_- \cdot (-\mathsf{Costs} - \mathsf{Penalties}_-) + (1-q_-) \cdot (-\mathsf{Costs})] + \\ + p \cdot [q \cdot (-\mathsf{Costs} + \mathsf{Gains} - \mathsf{Penalties}) + (1-q) \cdot (-\mathsf{Costs} + \mathsf{Gains})] = \\ = -\mathsf{Costs} + p \cdot (\mathsf{Gains} - q \cdot \mathsf{Penalties}) - (1-p) \cdot q_- \cdot \mathsf{Penalties}_- \,.\end{aligned}$$

We denote the average penalty of an attacker in case the attack was successful by π and the average penalty in case the attack was not successful by π_-, i.e. $\pi = q \cdot \mathsf{Penalties}$ and $\pi_- = q_- \cdot \mathsf{Penalties}_-$. Thus we have

$$\mathsf{Outcome} = -\mathsf{Costs} + p \cdot (\mathsf{Gains} - \pi) - (1-p) \cdot \pi_- \quad . \tag{2}$$

The attack is unlikely if $\mathsf{Outcome} < 0$.

These concepts are illustrated in Example 2.

4 The Method

In this section, we describe the threat-tree based security evaluation method, which consists of two phases: (1) identifying the primary threats (ultimate goals for attackers) and (2) breaking complex attacks into simpler ones and computing the threat tree in order to determine the most profitable attack and to decide whether the attacker's outcome is positive. Note that there are alternative ways of describing the attacks, for example one may use the *attack simulation method* [15] in which all possible attack paths are generated first and only after that the most likely attacks are analyzed.

4.1 Primary Threats

The security analysis of a system begins with identifying the *primary threats* – events that directly cause losses. For example, *"software bugs in firewall"* is not a primary threat, whereas *"lost market share because of stolen IP"* is a primary threat because of direct (monetary) losses.

Definition 2. *A system is said to be* practically secure against rational attacks *if every primary threat is unlikely, i.e. non-profitable for attackers.*

Example 3 shows two examples of primary threats a Company may consider.

For a software-developing Company, the primary threats can be

- *Forestalling release* – a competitor, by using a stolen code or architecture, launches a similar product to the market before the Company does it. This causes a lost market share.
- *Competitive release* – a competitor, by using a stolen code or architecture, launches a similar service/product soon after the Company does it.

Example 3: Primary threats for a Company.

4.2 Tree Computations

If the set $\mathfrak{T}$ of primary threats is fixed, the second step of the analysis is to construct an *attack tree* for each primary threat $\mathcal{T} \in \mathfrak{T}$. This is done by a gradual refinement procedure where each primary or intermediate threat (or attack) is split into simpler sub-attacks until one reaches the level of atomic threats where it no more makes sense to split them any further. We distinguish between two kinds of splits: AND-split (where all the sub-attacks must be completer in order to carry out a higher-level attack) and OR-split (where only one sub-attack is sufficient).

As a result of the process, we want to be able to compare the game values of different attack scenarios. Thus, we must specify how to determine the necessary parameters throughout the computations. From equation (2) we see that the required parameters are Costs, p, Gains, π, and π_-. Almost all of them make sense for all nodes in the tree, with the notable exception of Gains. It is very hard to say which percentage of the desired result is obtained, if only some sub-attack is completed. Still, in order to perform the computations in intermediate nodes, we need some intermediate value of Gains as well. In this paper we will assume that this value is constant throughout the tree and that it is equal to the Gains obtained by the attacker if he is able to complete a primary threat attack.

For the leaf nodes (atomic attacks), the rest of the parameter values $(\mathsf{Costs}, p, \pi, \pi_-)$ are deduced by the experts from the assumptions about the real environment. For non-leaf nodes, this quadruple is computed based on the corresponding parameters of the child nodes. In addition to the parameters, the $\mathsf{Outcome}$ value is computed for all nodes by applying (2). The parameters of non-leaf nodes (in binary case) are computed as follows:

- For an OR-node with child nodes with parameters $(\mathsf{Costs}_i, p_i, \pi_i, \pi_{i-})$ $(i = 1, 2)$ the parameters $(\mathsf{Costs}, p, \pi, \pi_-)$ are computed as:

$$(\mathsf{Costs}, p, \pi, \pi_-) = \begin{cases} (\mathsf{Costs}_1, p_1, \pi_1, \pi_{1-}), \text{ if } \mathsf{Outcome}_1 > \mathsf{Outcome}_2 \\ (\mathsf{Costs}_2, p_2, \pi_2, \pi_{2-}), \text{ if } \mathsf{Outcome}_1 \leq \mathsf{Outcome}_2 \end{cases},$$

  where $\mathsf{Outcome}_i = -\mathsf{Costs}_i + p_i \cdot \mathsf{Gains} - p_i \cdot \pi_i - (1 - p_i) \cdot \pi_{i-}$ $(i = 1, 2)$.
- For a AND-node with child nodes with parameters $(\mathsf{Costs}_i, p_i, \pi_i, \pi_{i-})$ $(i = 1, 2)$ the parameters $(\mathsf{Costs}, p, \pi, \pi_-)$ are computed as follows:

$$\mathsf{Costs} = \mathsf{Costs}_1 + \mathsf{Costs}_2, \quad p = p_1 \cdot p_2, \quad \pi = \pi_1 + \pi_2,$$
$$\pi_- = \frac{p_1(1 - p_2)(\pi_1 + \pi_{2-}) + (1 - p_1)p_2(\pi_{1-} + \pi_2) + (1 - p_1)(1 - p_2)(\pi_{1-} + \pi_{2-})}{1 - p_1 p_2}.$$

The formula for π_- represents the average penalty of an attacker, assuming that at least one of the two child-attacks was not successful. For example, if the first attack was successful and the second one unsuccessful (which is an event with probability $p_1(1 - p_2)$), then the average penalty of the attacker is $\pi_1 + \pi_{2-}$. Note that the formulae above have obvious generalizations for non-binary trees.

4.3 Example

We illustrate the computations by using the simplified threat three depicted in Figure 1. We assume that the profit obtained by the attacker by launching a Forestalling Release (i.e. the value of the Gains parameter) is $\$6,000,000$. The Company estimates the parameters of atomic threats as follows:

Stolen code is used in products. We assume that the cost of creating a product from a stolen code is about $\$10^6$. The success probability of the product creation process is estimated to 0.9. If the product creation is successful, then with probability $\frac{1}{6}$ the use of stolen code is detected and proved in court. The penalties in this case would be about $\$6,000,000$, thus $\pi = \frac{1}{6} \cdot \$6,000,000 = \$10^6$. If the project is not successful, then of course no damage is done to the Company and the attack will not be detected (at least with high probability), hence we take $\pi_- = 0$.

Bribe a programmer. We assume that about $\frac{1}{10}$ of the people can be bribed for 1 Million dollars. Hence, for the bribery, we take $\mathsf{Costs} = 10^6$ and $p = 0.1$. Briberies can be made anonymous by using a chain of middle-men. Hence, we assume that the probability of getting caught is quite low – about 0.001 – but it would still be reasonable to assume that in case the attacker (i.e. the competitor) is caught, the penalties are quite high – about 10^6. This includes the direct penalties and the loss of trust. We also assume that the probability of getting caught and the penalties for bribery do not depend on whether the bribery was successful. Hence, $\pi = \pi_- = 10^3$.

The results are summarized in Table 1.

Table 1. Computing the threat three of Figure 1.

	Description of threat	Type	Costs	p	π	π_-	Outcome
	Forestalling release	AND	$\mathbf{1,101,000}$	$\mathbf{0.405}$	$\mathbf{1,110,000}$	$\mathbf{941,933}$	$\mathbf{+319,000}$
B	Stolen code is used in products		10^6	0.9	$1 \cdot 10^6$	0	
A	Steal the code	OR	$\mathbf{101,000}$	$\mathbf{0.45}$	$\mathbf{110,000}$	$\mathbf{110,000}$	
A1	Get code by bribing a programmer	AND	$\mathbf{1,000,000}$	$\mathbf{0.09}$	$\mathbf{101,000}$	$\mathbf{101,000}$	$\mathbf{-561,000}$
A1.1	Bribe a programmer		10^6	0.1	10^3	10^3	
A1.2	Programmer obtains the code		0	0.9	10^5	10^5	
A2	Get code via network attack	AND	$\mathbf{11,000}$	$\mathbf{0.0027}$	$\mathbf{1,001}$	$\mathbf{911}$	$\mathbf{+4,289}$
A2.1	Employ a hacker		10^4	0.9	10^3	10^2	
A2.2	Hacker exploits a bug		10^3	0.5	1	1	
A2.1	There is a bug to exploit		0	0.006	0	0	
A3	Get code via physical robbery	AND	$\mathbf{101,000}$	$\mathbf{0.45}$	$\mathbf{110,000}$	$\mathbf{110,000}$	$\underline{\mathbf{+2,489,000}}$
A3.1	Employ a robber		10^5	0.9	10^4	10^4	
A3.2	Robber breaks into the system and obtains the code		10^3	0.5	10^5	10^5	

Programmer obtains the code. We assume that the internal security measures in the Company against stealing the code are not very efficient, so that about $\frac{1}{10}$ of the stealing attempts are detected. We believe that the real situation is much worse in most of the companies. If the programmer gets caught then the losses concern the loss of trust, i.e. it would be very difficult for the programmer to find job after such an incident. We estimate the losses of the programmer to be about $\$10^6$.

The parameters of other atomic attacks (A2.1–A3.2) should also be estimated (in a similar way) but we omit the reasoning about them in this paper. Note also that none of these numbers are results of rigorous (social) studies, but rather depend on the estimates given by the security expert. However, often giving estimates as bounds with the precision of order of magnitude is

quite enough. For example, in order to apply the OR-rule from Subsection 4.2, we only need to know which child node has the largest Outcome and just having some reasonable bounds is usually sufficient to take such decisions.

The results of the computations are presented in Table 1. First, we compute the sub-attacks A1, A2, and A3 (using the AND-node rule given in Subsection 4.2). We see that A3 is the attack with the highest Outcome and hence, by using the OR-node rule, we get the parameters for sub-attack A. Finally, by applying the AND-rule to A and B, we obtain the parameters of the "Forestalling release" attack. It turns out that

- the average outcome of the attacker is positive: $\mathsf{Outcome} = \$319,000$,
- the most profitable attack is a physical robbery.

The Company concludes that the system is insufficiently protected and measures must be taken against physical robbery. In the next section, we discuss how to decide about security measures in a rational way.

5 Security Measures

Rational choice of security measures is of the same importance as the estimation of risks. In this section, we recall the main types of measures and then introduce a simple metrics for economic justification of the measures. We also continue with our example in order to illustrate how the metrics works.

5.1 Types of Measures

To protect the system against attacks, various *security measures* can be taken. There are three main types of security measures:

- *Prevention measures* the purpose of which is to *reduce the success probability* p of attacks and to *increase the cost* of attacks. Physical access control mechanisms, suitable choice of information-transfer protocols, as well as properly stated home rules in the company are prevention measures.
- *Detection measures* the purpose of which is to *detect* the attack as fast as possible, and to *increase the probability* q *of getting caught*. Regular observation of competitors' business activities (e.g. in order to detect unfair use of stolen intellectual property), secure log mechanisms, patent protection of technical and business ideas (it helps to detect and prove unfair use of stolen information) – all these are detection measures, at least in the context of this work.
- *Recovery measures* the purpose of which is to *re-establish the normal functionality* of the system after an attack. Regular backups, insurance, etc. are recovery measures.

5.2 Rational Choice of Measures

The main practical questions about the security (expected to be answered during the security analysis) are the following:

- Are the current security measures *sufficient* to make the attacks non-profitable to attackers?
- Are the security measures *economically justified* (*necessary*), i.e. is their cost worth the risk they reduce? Security measures are never for free. It is hence reasonable to ask whether the additional level of security they offer is worth their price.

A software-developing Company may estimate the losses as follows: $\mathsf{Loss}[\text{"Forestalling release"}] = \$6,000,000$, $\mathsf{Loss}[\text{"Competitive release"}] = \$2,000,000$.

Example 4: Losses of the Company.

Let $\mathcal{M}$ denote the set of measures used in the Company and $\mathfrak{T}$ denote the set of primary threats. For each primary threat $\mathcal{T} \in \mathfrak{T}$, let $\mathsf{Loss}[\mathcal{T}]$ denote the losses associated with $\mathcal{T}$ as in Section 1 (see Example 4).

Let $\mathsf{Outcome}[\mathcal{T}]$ denote the outcome of the corresponding attack game for the attacker estimated by using attack trees. We define *total loss* – the largest potential loss caused by primary threats – as follows:

$$\mathsf{Loss}[\mathfrak{T}] = \max\{\mathsf{Loss}[\mathcal{T}]\colon\ \mathcal{T} \in \mathfrak{T},\ \text{and } \mathsf{Outcome}[\mathcal{T}] > 0\}\ .$$

In case the set of likely threats (those with $\mathsf{Outcome} > 0$) is empty, we set $\mathsf{Loss}[\mathfrak{T}] = 0$.

Let $\mathsf{Outcome}[\mathcal{T} \mid \mathcal{M}]$ denote the outcome of the attack game assuming that a set $\mathcal{M}$ of measures is taken in the system. The *conditional (total) loss* $\mathsf{Loss}[\mathfrak{T} \mid \mathcal{M}]$ is defined as follows:

$$\mathsf{Loss}[\mathfrak{T} \mid \mathcal{M}] = \max\{\mathsf{Loss}[\mathcal{T}]\colon\ \mathcal{T} \in \mathfrak{T},\ \text{and } \mathsf{Outcome}[\mathcal{T} \mid \mathcal{M}] > 0\}\ .$$

If for all primary threats $\mathcal{T}$, we have $\mathsf{Outcome}[\mathcal{T} \mid \mathcal{M}] \leq 0$, then no attack is profitable for the attacker, and we take $\mathsf{Loss}[\mathfrak{T} \mid \mathcal{M}] = 0$.

Definition 3. *A set $\mathcal{M}$ of measures is* sufficient (against rational attacks) *if* $\mathsf{Loss}[\mathfrak{T} \mid \mathcal{M}] = 0$. *A set $\mathcal{M}$ of measures is* adequate *(worth its cost) if* $\mathsf{Loss}[\mathfrak{T}] - \mathsf{Loss}[\mathfrak{T} \mid \mathcal{M}] > \mathsf{Cost}[\mathcal{M}]$.

Note that every adequate measure should make at least one primary attack unlikely. It is not sufficient for the adequacy that the average outcome of the attacker is diminished.

Definition 4. *A set $\mathcal{M}$ of measures is* (locally) optimal *if and only if it is sufficient, adequate, and* $\mathsf{Loss}[\mathfrak{T} \mid \mathcal{N}] > 0$ *for every proper subset $\mathcal{N} \subset \mathcal{M}$, i.e. no proper subset of $\mathcal{M}$ is sufficient.*

For a sufficient and adequate set $\mathcal{M}$ of measures we also have $\mathsf{Loss}[\mathfrak{T}] > \mathsf{Cost}[\mathcal{M}]$, leading us to a well-known conclusion that price of the defense measures should not exceed the value of the assets protected.

5.3 Example

We continue with the software-developing Company example. The conclusion of the risk analysis was that some additional physical protection mechanisms must be introduced in order to protect the Company against physical robbery. Say we have two offers to the Company from security companies with the following parameters:

- Company X offers a protection package $\mathcal{M}_X$ with price $\mathsf{Cost}[\mathcal{M}_X] = \$2,000,000$. The package is oriented to physical protection and reduces the probability that a robber breaks into the system from 0.5 to 0.25.
- Company Y offers a protection package $\mathcal{M}_Y$ with price $\mathsf{Cost}[\mathcal{M}_Y] = \$1,000,000$. The package is oriented to detection measures and increases twice the detection probabilities q and q_-, which means that also the average penalties π and π_- are increased twice.

Table 2. Computing the threat three for the system protected with $\mathcal{M}_X$.

	Description of threat	Type	Costs	p	π	π_-	Outcome
	Forestalling release	AND	$\mathbf{1,101,000}$	$\mathbf{0.2025}$	$\mathbf{1,110,000}$	$\mathbf{984,608}$	$\mathbf{-896,000}$
B	Stolen code is used in products		10^6	0.9	$1 \cdot 10^6$	0	
A	Steal the code	OR	$\mathbf{101,000}$	$\mathbf{0.225}$	$\mathbf{110,000}$	$\mathbf{110,000}$	
A1	Get code by bribing a programmer	AND	$\mathbf{1,000,000}$	$\mathbf{0.09}$	$\mathbf{101,000}$	$\mathbf{101,000}$	$\mathbf{-561,000}$
A2	Get code via network attack	AND	$\mathbf{11,000}$	$\mathbf{0.0027}$	$\mathbf{1,001}$	$\mathbf{911}$	$\mathbf{+4,289}$
A3	Get code via physical robbery	AND	$\mathbf{101,000}$	$\mathbf{0.225}$	$\mathbf{110,000}$	$\mathbf{110,000}$	$\mathbf{+1,139,000}$
A3.1	Employ a robber		10^5	0.9	10^4	10^4	
A3.2	Robber breaks into the system and obtains the code		10^3	0.25	10^5	10^5	

Which package to choose? If both packages are adequate (i.e. make the "Forestalling release" threat unlikely), then it is reasonable to choose $\mathcal{M}_Y$ because of lower price. If one of the packages turns out to be inadequate, then this package cannot be chosen, regardless of the price. Hence, it remains to determine whether the packages are adequate. We start from $\mathcal{M}_X$. The computations are shown in Table 2.

We see that $\mathcal{M}_X$ is sufficient as it makes the attack unlikely. It is also adequate since its cost was $\$2,000,000$, but the prevented loss was $\$6,000,000$.

Now we do the same with the package $\mathcal{M}_Y$. The results are presented in Table 3.

Table 3. Computing the threat three for the system protected with $\mathcal{M}_Y$.

	Description of threat	Type	Costs	p	π	π_-	Outcome
	Forestalling release	AND	$\mathbf{1,101,000}$	$\mathbf{0.405}$	$\mathbf{1,210,000}$	$\mathbf{1,041,933}$	$\mathbf{+219,000}$
B	Stolen code is used in products		10^6	0.9	$1 \cdot 10^6$	0	
A	Steal the code	OR	$\mathbf{101,000}$	$\mathbf{0.45}$	$\mathbf{210,000}$	$\mathbf{210,000}$	
A1	Get code by bribing a programmer	AND	$\mathbf{1,000,000}$	$\mathbf{0.09}$	$\mathbf{101,000}$	$\mathbf{101,000}$	$\mathbf{-561,000}$
A2	Get code via network attack	AND	$\mathbf{11,000}$	$\mathbf{0.0027}$	$\mathbf{1,001}$	$\mathbf{911}$	$\mathbf{+4,289}$
A3	Get code via physical robbery	AND	$\mathbf{101,000}$	$\mathbf{0.45}$	$\mathbf{210,000}$	$\mathbf{210,000}$	$\mathbf{+2,389,000}$
A3.1	Employ a robber		10^5	0.9	10^4	10^4	
A3.2	Robber breaks into the system and obtains the code		10^3	0.5	$2 \cdot 10^5$	$2 \cdot 10^5$	

As we can see, $\mathcal{M}_Y$ is not even sufficient and the attack is still likely. Hence, it is reasonable to buy the package $\mathcal{M}_X$, in spite of its higher price.

6 Conclusions and Further Work

We have used the presented simple risk-analysis framework several times in practice and have found it to be very suitable. The main benefits of the framework are that (1) it provides a systematic approach to the whole security analysis task and avoids a risk-analyst from getting lost in unimportant (technical) details, (2) it is easy to implement in a computer, (3) the main principles of the method are easily understandable to the people who make financial decisions and hence it can be used to justify investments into security.

It may seem that the method uses many unknown parameters like "the sum of money needed for bribing an employee" etc. At the same time, *these parameters are essential in any other risk analysis method* that is claimed to be adequate. Our method (and the threat-tree method in general) helps to determine systematically the (social) parameters we need to know for practical security estimation, and this is, in turn, an advantage of the method.

There are still several things to be improved in the method:

- Gains is a global parameter in the whole threat-tree and is used to make decisions in all OR-nodes. This makes the computations "greedy", i.e. the complexity is linear in the number of nodes. The "local" decisions made separately in OR-nodes not necessarily give the successful attack with the highest outcome. In order to get the global maximum, we have to examine all combinations of decisions in all OR-nodes. For example, if the tree contains m binary OR-nodes, the whole tree has to be computed 2^m times. It is not yet known how much effect this would give in practical threat trees. Further, it is possible to extend the model considering different amounts of Gains depending on whether the attack was successful or not.
- We assumed that all atomic attacks (or at least all children of AND-nodes) are independent of each other. This may not be the case. It seems that in practical security analysis we can build the tree so that possible dependencies do not have an effect. However, it is not excluded that in some cases we cannot avoid the dependencies. This needs some further research.

Risk analysis methods have not yet been discussed extensively in academic papers. In our opinion, one of the reasons has been that many such methods were (and are) business secrets of risk analysis companies. Considering the latest trends that computer criminals co-operate (and compete!) intensively, it seems to be the right time to start intense academic cooperation on the general risk analysis issues.

References

1. Daniel Geer, Kevin Soo Hoo, and Andrew Jaquith. Information security: Why the future belongs to the quants. *IEEE Security and Privacy*, 1(4):24–32, 2003.
2. Wes Sonnenreich, Jason Albanese, and Bruce Stout. Return On Security Investment (ROSI) – A practical quantitative model. *Journal of Research and Practice in Information Technology*, 38(1):55–66, February 2006.
3. Yvo Desmedt. Potential impacts of a growing gap between theory and practice in information security. In *Information Security and Privacy: 10th Australasian Conference, ACISP 2005*, LNCS 3524, pages 532–536. Springer, 2005.
4. James W. Meritt. A method for quantitative risk analysis. In *Proceedings of the 22nd National Information Systems Security Conference*, 1999.
5. W.E. Vesely, F.F. Goldberg, N.H. Roberts, and D.F. Haasl. *Fault Tree Handbook*. US Government Printing Office, January 1981. Systems and Reliability Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission.
6. John Viega and Gary McGraw. *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison Wesley Professional, 2001.
7. Andrew P. Moore, Robert J. Ellison, and Richard C. Linger. Attack modeling for information security and survivability. Technical Report CMU/SEI-2001-TN-001, Software Engineering Institute, 2001.
8. Bruce Schneier. Attack trees: Modeling security threats. *Dr. Dobb's Journal*, 24(12):21–29, December 1999.
9. Bruce Schneier. *Secrets & Lies. Digital Security in a Networked World*. John Wiley & Sons, 2000.
10. Sjouke Mauw and Martijn Oostdijk. Foundations of attack trees. In Dongho Won and Seungjoo Kim, editors, *International Conference on Information Security and Cryptology – ICISC 2005*, LNCS 3935, pages 186–198. Springer, December 2005. In print.
11. Alexander Opel. Design and implementation of a support tool for attack trees. Technical report, Otto-von-Guericke University, March 2005. Internship Thesis.
12. Peng Liu, Wanyu Zang, and Meng Yu. Incentive-Based Modeling and Inference of Attacker Intent, Objectives and Strategies. *ACM Transactions on Information and Systems Security*, 8(1):78 –118, 2005.
13. Stuart E. Schechter. *Computer Security Strength & Risk: A Quantitative Approach*. PhD thesis, Harvard University, 2004.
14. 2004 E-CrimeWatch Survey. Summary of Findings. Conducted by CSO magazine in cooperation with the U.S. Secret Service & CERT Coordination Center. Available at `http://www.cert.org/archive/pdf/2004eCrimeWatchSummary.pdf`, 2004.
15. Gidi Cohen. The role of attack simulation in automating security risk management. *Information Systems Control Journal*, 1:51–54, 2005.