Quantitative Analysis of Attack Defense Trees

Nihal Pekergin(✉) and Sovanna Tan

LACL, Univ Paris Est Créteil, 94010 Créteil, France
{nihal.pekergin,sovanna.tan}@u-pec.fr

Abstract. The quantitative analysis of Attack Tree models brings insights on the underlying security-critical systems. Having information on temporal behaviours of such systems lets us check whether at a given time, the probability that the system is compromised is less than a critical threshold or not. Moreover the evaluation of the countermeasure efficiency and the determination of eventual reinforcements of security-critical systems are very important. In this paper, we extend the approach proposed in [11] for numerical analysis of the Attack Tree models to the Attack Defense Tree analysis. The completion times of attacks and countermeasures are defined by finite discrete random variables. The output distribution of the root of an Attack Defense Tree is computed by a bottom-up approach. However the size of the output distribution can become quickly very large. We prove that the method which consists in deriving bounding distributions of reduced sizes by means of the stochastic comparison method can be used in the presence of counter-measure gates.

Keywords: Attack defense tree · Discrete probability distribution · Stochastic bounds

1 Introduction

The graphical formalisms such as Fault Trees (FT) [14] and Attack Trees(AT) are commonly used models for safety and security analysis. In Attack Trees the leaves are the actions taken intentionally by attackers called basic attacks. A leaf turns out $True$ when the underlying attack is successful. The complex attack scenarios may be specified by combining basic attacks with logical gates. The output of a gate is $True$ if the subsystem having this gate as root is compromised. Thus the root of the AT turns out $True$ when the whole system is compromised.

The Attack Trees were first proposed in [15]. Recently, several works have been done in the literature (see [8,17] and the references therein). These works can be classified as semantical approaches to give a rigorous, mathematical definition of AT trees and their extensions; generation approaches to study the construction of such models; quantitative approaches to propose efficient algorithms and techniques for the quantitative analysis.

Q. Zhao and L. Xia (Eds.): VALUETOOLS 2021, LNICST 404, pp. 207–220, 2021.
https://doi.org/10.1007/978-3-030-92511-6_13

In the first AT models the logical connectors were limited to the AND and OR gates, but they have been extended by including some dynamical gates to the Attack-Fault Trees [10], and by adding countermeasures [13,17] to the Attack-Defense Trees. For the static quantitative evaluation of such models, the leaves (basic events) are specified by the success probability of the underlying event. By applying the standard bottom-up algorithm, the success probability of attack scenarios specified by an AT can be computed [13]. Temporal behaviours of safety-security critical systems are primordial. For instance, the time for an attack to be successful is an important indicator of the safety properties. Thus, it is important to check whether at a given time the probability that the system is compromised is not greater than a critical threshold.

In [2], the authors propose to specify the time to success of an attack by *Acyclic Phase Type*(APH) distributions. The distributions associated with leaves may be any continuous distribution since fitting algorithms exist in the literature to approximate a continuous distribution by a APH distribution. By considering independence of basic attacks, the random variables of the gate outputs can be computed as the maximum (AND gate), the minimum (OR gate) and convolution of (SEQ gate) of the underlying input distributions. However the successive applications of these operators may lead to a state space explosion of the output distributions. To overcome this problem, the output distributions are *compressed* to construct approximate, smaller sizes APH distributions by algorithms of cubic computational complexity.

In [11] we have proposed an approach to quantitatively analyze AT based on ideas similar to those exposed in [2] but with radically different techniques. The temporal behaviors of basic events are specified by discrete probability distributions. We prevent the size explosion problem by replacing the output distribution with reduced-size bounding distributions in the sense of the strong stochastic ordering ($\leq_{st}$). Intuitively speaking, if two distributions are ordered in the sense of this order: $d_1 \leq_{st} d_2$, then the cumulative probability distribution of d_1 is always greater or equal to the cumulative probability distribution of d_2 (d_2 takes larger values than d_1). In other words given a time (t), in the upper bounding distribution the probability that the time to success is greater or equal to t is greater or equal to the probability computed by the original distribution.

The lower and upper reduced-size bounding distributions can be derived due to the monotonicity properties of gates (AND, OR, SEQ). Roughly speaking, monotonicity can be explained with the following property: if the input distributions are replaced by upper (resp. lower) bounding distributions the output distribution is also a upper (resp. lower) bounding distribution. Indeed these gates satisfy the monotonicity in the sense of the $\leq_{st}$ ordering since the related operations are non decreasing functions on the inputs.

The size reduction can be performed with several compressing algorithms having different computational complexities. The naive algorithm has linear complexity. It consists in merging the successive atoms and in putting the sum of corresponding probabilities to the largest value for the upper bound and to the smallest one for the lower bound. In this framework, it is possible to consider continuous distributions for the times of events and then to apply bounding discretization In [1], the algorithms to construct a bounding distribution in the

sense of the $\leq_{st}$ ordering of size K for an input distribution of size N, with $K \leq N$ are given. The optimal algorithm with respect to a given non decreasing reward has $O(KN^2)$ computational complexity, but several greedy algorithms with lower computational complexities have been also presented.

In this paper, we propose to extend this approach to evaluate the countermeasures associated with AND gates as proposed in [7,13]. We first explain how the countermeasure nodes which are included in AT models to counter attacks can be considered in our framework. The times for the attack and the countermeasure to succeed have contrary impacts on the whole temporal behaviours of the underlying Attack Defense Tree. Roughly speaking if the time needed to complete a countermeasure decreases, the attack takes longer to succeed. Thus the underlying operation does not have non decreasing monotonicity property and its analysis needs some attention. We first show that adding a countermeasure input to an AND gate does not increase the size of the output distribution. We propose not to compress the countermeasure node (leaf) and show that it is still possible to compress other distributions.

These last years, many works to enhance the design and analysis of attack-defense trees with formal methods such as model checking, automata theory, constraint solving, Bayesian networks have been done. The stochastic game interpretation of Attack-Defense trees and its analysis with PRISM-Games model checker is given in [3]. The priced-timed automata modelling of ATs and analysis with UPPAAL model checker can be found in [5,9]. The extended asynchronous multi-agent systems formalism has been proposed in [12]. These works show the increasing interest of the community for the quantitative analysis of such models.

The paper is organized as follows. In Sect. 2 we present the analysis of ATs with the countermeasure nodes. We illustrate this approach with an example of the literature in Sect. 3. Finally we conclude and give perspectives.

2 Quantitative Evaluation of Attack Defense Trees

2.1 Bounding Distributions for Attack Trees

We have proposed in [11], the quantitative evaluation of Attack Trees with conjunction (AND), disjunction (OR) and sequential conjunction (noted as SEQ or $SAND$). A finite discrete probability distribution is associated with each Basic Attack (BA). This random variable is indeed the time at which the corresponding BA occurs. The input of a gate associated with this distribution turns out ($True$) at this time. Similarly, the output distribution of a gate corresponds to the time at which the subsystem having this gate as root is compromised. Therefore the output distribution of the root of an Attack Tree denotes the time when the whole system is compromised. The model is observed during time interval $0 \leq t < MT$. The time value MT represents indeed the infinity such that events which occur at this date do not indeed happen. The output distribution of the formerly stated gates can be defined as a function of input distributions [2,11]. Let I_1 I_2 be the input distributions. Under independent basic attacks and discrete distributions, the output distributions are numerically computed as:

- $\mathcal{P}(O_{OR}(I_1, I_2) = a)$
  $= \mathcal{P}(I_1 = a).\mathcal{P}(I_2 > a) + \mathcal{P}(I_2 = a).\mathcal{P}(I_1 > a) + \mathcal{P}(I_1 = a) \times \mathcal{P}(I_2 = a)$
- $\mathcal{P}(O_{AND}(I_1, I_2) = a)$
  $= \mathcal{P}(I_1 = a).\mathcal{P}(I_2 < a) + \mathcal{P}(I_2 = a).\mathcal{P}(I_1 < a) + \mathcal{P}(I_1 = a) \times \mathcal{P}(I_2 = a)$
- $\mathcal{P}(O_{SEQ}(I_1, I_2) = a) = \sum_k \mathcal{P}(I_1 = k) \times \mathcal{P}(I_2 = (a - k))$

Thanks to the tree structure of the model, the output distributions of the gates and finally the output distribution of the root (the time for the system to be compromised) is computed by the bottom-up approach.

The sizes of distributions (the number of atoms) can increase rapidly due to the successive applications of the operations associated with the gates. For AND, OR gates, the size of the output distribution may be the sum of the input distribution sizes, $O(l_1 + l_2)$. However for the convolution operation, the number of atoms may be the product of the input distribution sizes, $(O(l_1 \times l_2))$. The main drawback of this approach is the explosion of the number of atoms after successive computations. We have proposed to reduce the number of atoms and construct bounding distributions by compressing [11]. The computation complexity of the output distribution depends on the sizes of the input distributions. Let l_1, l_2 be two input distributions sizes and $l = \max(l_1, l_2)$. The output distributions of the above gates can be computed by a naive algorithm with complexity $\Theta(l_1 \times l_2)$. The computation complexities are indeed bounded by $\Theta(l \times \log l)$ (Discrete Fast Fourier transform for convolution and sorting for other gates). Obviously, the size reduction will be favoring for computation complexities and also for the numerical stability since we consider probability vectors.

The ability to control the distribution sizes during the bottom-up analysis of an AT is of great importance for the point of view of the algorithmic complexity. In [4,11], we have shown that the reduced-size bounding output distributions can be derived by considering bounding input distributions for AND, OR, SEQ gates. These bounding distributions are in the sense of the stochastic strong order ($\leq_{st}$) associated with non decreasing functions. Thus if two random variables are ordered in the $\leq_{st}$ order then their non decreasing rewards are also ordered:

$$A \leq_{st} A^u \quad \Leftrightarrow \quad \mathrm{E}[f(A)] \leq \mathrm{E}[f(A^u)] \tag{1}$$

for all non decreasing function f, when the expectations E exist. For instance, if $A \leq_{st} A^u$, then $\mathrm{E}[A] \leq \mathrm{E}[A^u]$.

The monotonicity properties of AND, OR, SEQ are proved in [4,11]. Let I_1, I_2 be the input distributions and $GATE(I_1, I_2)$ be the output distribution of a AND, OR, SEQ gate. The output of the the gate is upper (resp. lower) bounded if it is subjected to upper (resp. lower) bounded input distributions:

Property 1.

$$I_1 \leq_{st} I_1^{up}, I_2 \leq_{st} I_2^{up} \Rightarrow GATE(I_1, I_2) \leq_{st} GATE(I_1^{up}, I_2^{up})$$

The monotonicity results from the fact that the output of these gates are non decreasing functions of their inputs. This monotonicity property lets us construct the reduced-size distributions and diminish the algorithmic complexity.

The reduced-size bounding distributions in the sense of the $\leq_{st}$ ordering can be constructed in several manners. The naive approach to construct an upper bounding (resp. lower) distribution to divide the size by m is to take m successive values, to delete the smallest (resp. greatest) one and to include its probability to the greatest (resp. smallest) value [16]. In [1], an algorithm to construct for an arbitrary distribution of size N bounding distributions of size $K < N$ is proposed. This algorithm is optimal such that it is the smallest (resp. greatest) $\leq_{st}$ (Eq. (1)) upper (resp. lower) bound respect to a non decreasing reward (for example expectation). is given: and any positive non decreasing reward function , This optimal algorithm is based on a graph optimization problem and can be computed by a dynamical programming approach with $\Theta(KN^2)$. Greedy algorithms with less computational complexities have been also proposed. The sizes of the bounding distributions K can be reduced decrementally by taking care of the accuracy and the computational complexity trade-off.

2.2 *SI-AND* Gate with Countermeasure

Attack Trees are extended to Attack-Defense Trees (ADT) with two actors: attacker and defender. The goal of each actor is to counter the other one. We consider the semantic defined in [7,13] for the countermeasure nodes (defenders). Let A and D be respectively discrete distributions representing the time for an attack to be successful and the time to a countermeasure to be operational. The output of this gate turns out $True$ if the input of the attack A is $True$ and the input of the countermeasure is not yet operational thus $False$. Let O be the output distribution of a $SI\text{-}AND(A, D)$ gate having attack A and countermeasure D as inputs. The output distribution of this gate with mutually independent inputs is computed as:

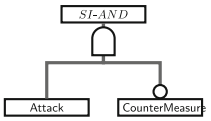

Fig. 1. *SI-AND* gate with a single inverter on countermeasure input

$$\mathcal{P}(O = i) = \mathcal{P}(A = i).\mathcal{P}(D > i), \quad i \neq MT \tag{2}$$

$$\mathcal{P}(O = MT) = 1 - \sum_{j|j \neq MT} \mathcal{P}(O = j) \tag{3}$$

Eq. (2) means that attack at time i is successful, if the countermeasure is not operational at time i. The probability of this event is the product of the probability to have an attack at time i and the probability that the countermeasure is not operational at time i which means that the countermeasure becomes operational at a time later than i. Equation (3) is derived since the probabilities are

summed up to 1. This is the probability that the output does not turn out $True$ (the probability that the attack will not happen) during the observation time interval $[0 - MT[$.

The temporal properties of this gate with discrete random inputs A, D specified by their state space, $\mathcal{V}$ and the probability vectors, $\mathcal{P}$ are given in Table 1. The state space of the output, $\mathcal{V}_O$ will be the same as the state space of the attack, $\mathcal{V}_A$. The probability that A is successful at time 2 is always 0.25, since the earliest time the countermeasure is operational is time 4. Thus the attack at time 2 can not be countered. The probability that A is successful at time 5 is the product of the probability that attack is successful at time 5 and the probability that the countermeasure is operational later than time 5, so it is $0.2.(1 - 0.15) = 0.17$.

Table 1. Input and output distributions for the gate given in Fig. 1

Attack Dist.					
$\mathcal{V}_A$	2	5	8.5	10	15
$\mathcal{P}_A$	0.25	0.2	0.3	0.15	0.1

Bound on Attack Dist.			
$\mathcal{V}_{A^u}$	4	10	15
$\mathcal{P}_{A^u}$	0.25	0.65	0.1

Defense Dist.			
$\mathcal{V}_D$	4	7	15
$\mathcal{P}_D$	0.15	0.4	0.45

Output Dist.					
$\mathcal{V}_O$	2	5	8.5	10	15
$\mathcal{P}_O$	0.25	0.17	0.135	0.0675	0.3775

Bound on Output Dist.			
$\mathcal{V}_{O^u}$	4	10	15
$\mathcal{P}_{O^u}$	0.2125	0.2925	0.495

2.3 Monotonicity Properties

Taking benefice of the monotonicity of a $SI\text{-}AND$ gate with countermeasure D is more subtle since the two inputs of this gate have contrary roles. We assume that the leaves which are countermeasure nodes are not compressed. Notice that we do not need to compress the output of such gates since the output distribution has the same state space as its attack input whatever the state space of the countermeasure input is. Therefore there is no explosion of the state space of the output due to the countermeasure input. However the attack input may be a bounding distribution of a subsystem. We must show that if the attack input is a bounding distribution, the output distribution of the $SI\text{-}AND$ gate will remain a bounding distribution.

We first illustrate this property with an example. First let us write $\leq_{st}$ inequalities (Eq. (1)) for $A \leq_{st} A^u$ where n_A denotes the size of the vectors and the indices of vectors are $1 \leq i \leq n_A$:

$$i \in \{n_A - 1, n_A - 2, \cdots, 2, 1\}, \quad \sum_{\{k|k>i\}} \mathcal{P}_A[k] \leq \sum_{\{j|\mathcal{V}_{A^u}[j]>\mathcal{V}_A[i]\}} \mathcal{P}_{A^u}[j].$$

Since the sum of probabilities is equal to 1, these inequalities can be written as follows:

$$i \in \{1, 2, \cdots, n_A - 2, n_A - 1\}, \quad \sum_{\{k|k\leq i\}} \mathcal{P}_A[k] \geq \sum_{\{j|\mathcal{V}_{A^u}[j]\leq\mathcal{V}_A[i]\}} \mathcal{P}_{A^u}[j].$$

The inequalities for A and the reduced size upper bound, A^u are

$$0.1 \leq 0.1;\ \ 0.25 \leq 0.75;\ \ 0.55 \leq 0.75;\ \ 0.75 \leq 0.75$$

The output distributions can be computed by Eqs. (2), (3) with $MT = 15$. The inequalities for $O \leq_{st} O^u$ are

$$0.3775 \leq 0.495;\ \ 0.0675{+}0.3775 \leq 0.2925 + 0.495;$$

$0.135{+}0.0675 + 0.3775 \leq 0.2925 + 0.495;\ \ 0.17 + 0.135 + 0.0675 + 0.3775 \leq 0.2925 + 0.495$.

Therefore we conclude that $O \leq_{st} O^u$.

Proposition 1. *We consider the output distributions for a SI-AND gate with countermeasure D. We note the output of the gate with attack input A and A^u as follows:*

$$O = \textit{SI-AND}(A, D), \quad O^u = \textit{SI-AND}(A^u, D)$$

We have the following property:

$$\textit{If } A \leq_{st} A^u \textit{ then } O \leq_{st} O^u.$$

Proof. For the sake of simplicity, we assume that A and A^u take values in the same set which is the union set of their respective state spaces: $\mathcal{V}_A \cup \mathcal{V}_{A^u}$. This is indeed the case if we put null probabilities when the random variable does not take the corresponding value. Let N be the size of the union set. The vectors corresponding to A and A^u are indexed by $1 \leq i \leq N$.

$$\mathcal{V}_{\mathcal{A}}[i] = \mathcal{V}_{A^u}[i],\ \ 1 \leq i \leq N \text{ and } \mathcal{V}_A[N] = \mathcal{V}_{A^u}[N] = MT.$$

With these notations, the inequalities for $A \leq_{st} A^u$ as :

$$\begin{aligned}
&i = 1, && \mathcal{P}_A[1] \geq \mathcal{P}_{A^u}[1] \\
&i = 2, && \mathcal{P}_A[1] + \mathcal{P}_A[2] \geq \mathcal{P}_{A^u}[1] + \mathcal{P}_{A^u}[2] \\
&\vdots && \\
&i = N-1, && \sum_{j=1}^{N-1} \mathcal{P}_A[j] \geq \sum_{j=1}^{N-1} \mathcal{P}_{A^u}[j] \\
&i = N, && \sum_{j=1}^{N} \mathcal{P}_A[j] = \sum_{j=1}^{N} \mathcal{P}_{A^u}[j] = 1
\end{aligned}$$

The $O \leq_{st} O^u$ inequalities that must be satisfied for the $\leq_{st}$ comparison on the output distributions of the *SI-AND* gate (Eq. (2) and (3)).

The first inequality, $i = 1$:

$$\mathcal{P}_A[1] \left(\sum_{\substack{k| \\ \mathcal{V}_D[k] > \mathcal{V}_A[1]}} \mathcal{P}_D[k] \right) \geq \mathcal{P}_{A^u}[1] \left(\sum_{\substack{k| \\ \mathcal{V}_D[k] > \mathcal{V}_A[1]}} P_D[k] \right)$$

This is indeed the first inequality for $A \leq_{st} A^u$ but multiplied in both parts by the same positive value ($\sum_{k|V_D[k]>V_A[1]}$). Thus the first inequality is satisfied.

The second inequality, $i = 2$:

$$\mathcal{P}_A[1]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]>\mathcal{V}_A[1]}} \mathcal{P}_D[k]\right) + \mathcal{P}_A[2]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]>\mathcal{V}_A[2]}} \mathcal{P}_D[k]\right)$$

$$\geq \mathcal{P}_{A^u}[1]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]>\mathcal{V}_A[1]}} \mathcal{P}_D[k]\right) + \mathcal{P}_{A^u}[2]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]>\mathcal{V}_A[2]}} \mathcal{P}_D[k]\right)$$

This can be written as follows:

$$(\mathcal{P}_A[1] + \mathcal{P}_A[2])\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]>\mathcal{V}_A[2]}} \mathcal{P}_D[k]\right) + \mathcal{P}_A[1]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]\leq\mathcal{V}_A[2]\\ \mathcal{V}_D[k]>\mathcal{V}_A[1]}} \mathcal{P}_D[k]\right)$$

$$\geq (\mathcal{P}_{A^u}[1] + \mathcal{P}_{A^u}[2])\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]>\mathcal{V}_A[2]}} \mathcal{P}_D[k]\right) + \mathcal{P}_{A^u}[1]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]\leq\mathcal{V}_A[2]\\ \mathcal{V}_D[k]>\mathcal{V}_A[1]}} \mathcal{P}_D[k]\right)$$

Similarly to the above case, this inequality can be derived from the two first inequalities for $A \leq_{st} A^u$ by multiplying both parts by the same values.

The inequality for $i = m$:

$$\sum_{j=1}^{m} \mathcal{P}_A[j]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]>\mathcal{V}_A[j]}} \mathcal{P}_D[k]\right) \geq \sum_{j=1}^{m} \mathcal{P}_{A^u}[j]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]>\mathcal{V}_A[j]}} \mathcal{P}_D[k]\right)$$

This inequality can be organized as follows:

$$\sum_{j=1}^{m} \mathcal{P}_A[j]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]>\mathcal{V}_A[m]}} \mathcal{P}_D[k]\right) + \sum_{j=1}^{m-1} \mathcal{P}_A[j]\left(\sum_{\substack{k|\\ \mathcal{V}_D[k]\leq\mathcal{V}_A[m]\\ \mathcal{V}_D[k]>\mathcal{V}_A[m-1]}} \mathcal{P}_D[k]\right) + \cdots$$

$$\cdots + \sum_{j=1}^{2} \mathcal{P}_A[j] \left(\sum_{\substack{k| \\ \mathcal{V}_D[k] \leq \mathcal{V}_A[3] \\ \mathcal{V}_D[k] > \mathcal{V}_A[2]}} \mathcal{P}_D[k] \right) + \mathcal{P}_A[1] \left(\sum_{\substack{k| \\ \mathcal{V}_D[k] \leq \mathcal{V}_A[2] \\ \mathcal{V}_D[k] > \mathcal{V}_A[1]}} \mathcal{P}_D[k] \right)$$

$$\geq \sum_{j=1}^{m} \mathcal{P}_{A^u}[j] \left(\sum_{\substack{k| \\ \mathcal{V}_D[k] > \mathcal{V}_A[m]}} \mathcal{P}_D[k] \right) + \sum_{j=1}^{m-1} \mathcal{P}_{A^u}[j] \left(\sum_{\substack{k| \\ \mathcal{V}_D[k] \leq \mathcal{V}_A[m] \\ \mathcal{V}_D[k] > \mathcal{V}_A[m-1]}} \mathcal{P}_D[k] \right) +$$

$$\cdots + \sum_{j=1}^{2} \mathcal{P}_{A^u}[j] \left(\sum_{\substack{k| \\ \mathcal{V}_D[k] \leq \mathcal{V}_A[3] \\ \mathcal{V}_D[k] > \mathcal{V}_A[2]}} \mathcal{P}_D[k] \right) + \mathcal{P}_{A^u}[1] \left(\sum_{\substack{k| \\ \mathcal{V}_D[k] \leq \mathcal{V}_A[2] \\ \mathcal{V}_D[k] > \mathcal{V}_A[1]}} \mathcal{P}_D[k] \right)$$

It can be seen that the inequality for $i = m$ can be derived from the m first inequalities for $A \leq_{st} A^u$ by multiplying both part by the same positive values. This completes the proof. □

This proposition shows that the monotonicity properties are satisfied to construct bounding output distributions, if the distributions at leaves associated with countermeasures are not compressed. Therefore it is possible to compress distributions to construct reduced-size output distributions of Attack-Defence models to overcome state space explosion problem. Remind that the countermeasures do not increase the size of the output distribution, thus there is no need apriori to compress discrete countermeasure distributions.

The presented gate specifications and the above algebraic proof are rather implementation oriented. Notice that these can be given with means of operations on input random variables, I_1, I_2 and the observation time MT:

$$\begin{aligned} AND(I_1, I_2) &= \min(MT, \max(I_1, I_2)) \\ OR((I_1, I_2) &= \min(MT, \min(I_1, I_2)) \\ SEQ(I_1, I_2) &= \min(MT, I_1 + I_2) \end{aligned}$$

As stated before, the monotonicity of these gates results from the generic definition of $\leq_{st}$ order (Eq. (1)) since the corresponding functions are non decreasing with respect to input parameters. The $SI\text{-}AND(A, D)$ gate with the observation time fixed to a constant value MT can be defined as:

$$SI\text{-}AND(A, D) = A \, \mathbb{1}_{A<\min(D,MT)} + MT \, \mathbb{1}_{A>=\min(D,MT)}$$

The left-side of the equation represents the case when the attack can take place while the right-side corresponds to the case when either the countermeasure

(defense) prevents the attacks or the observation time is over. The above monotonicity proof of this gate can be also directly established since this gate is defined as a non decreasing function with respect to A. The opposite impacts of attack and defense inputs can be seen in the above equation, since the output is a non increasing function with respect to D while it is non decreasing with respect to A.

In the case where distributions are continuous, the proposed methodology can be started with constructing bounding discrete distributions. In such cases, since attacks and defenses have contrary roles, if one aims to construct upper (resp. lower) bounds on the overall Attack Defense trees, upper bounds (lower) must be considered for the attacks while lower (upper) bounds must be considered for the defenses.

3 Case Studies

In this section we consider a case study from [6] in which temporal behaviours of attack defense trees are studied through the underlying Markov chain. The attack tree is given in Fig. 2. This example has three sequential steps. An attacker wants to compromise a server by *running malicious scripts* on it. To do so, they must first identify the target network address by *scanning the network*. Then they have to gain root access to run the malicious scripts. Thus the root of the attack tree is a *SEQ* gate. To prevent network address discovery, a *mutable network* with a dynamic topology reconfiguration is used as a countermeasure. This is represented with a *SI-AND* gate (*target identified*). There is an alternative to perform the second step, the *privilege escalation* : either to *use a security*

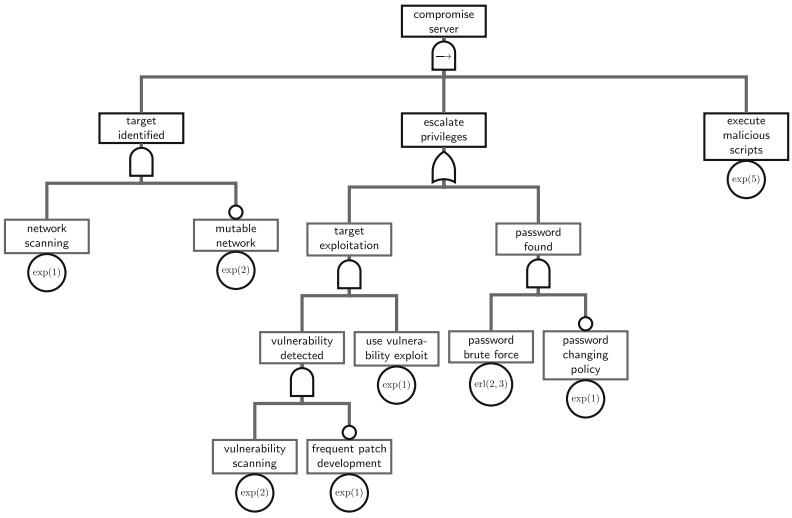

Fig. 2. Attack Tree *Compromise Server*.

vulnerability of the target or to *guess its root password*. It is represented with an *OR* gate. To do the *target exploitation*, the attacker must *scan the target vulnerabilities*. *Frequent security patch development* and frequent updates are used as countermeasure. This is materialized by a *SI-AND* gate. Once a vulnerability has been discovered, the ability to use it, called *use vulnerability exploit* on the AT, is needed. This is represented by *AND* gate. Another countermeasure, a *password changing policy* is used to prevent the *password brute force attack* represented with a third *SI-AND* gate. In [6], completion times of basic attacks and countermeasures have been taken as truncated exponential and Erlang distributions. The *network scanning* basic attack is an exponential of rate 1. The *mutable network* defence is an exponential of rate 2. The *vulnerability scanning* attack is an exponential of rate 2. The *frequent patch development* defence is an exponential of rate 1. The *use vulnerability exploit* is an exponential of rate 1. The *password brute force* attack is an Erlang distribution with shape 2 and rate 3. The *password changing policy* defence is an exponential of rate 1. The *execute malicious scripts* basic attack is an exponential of rate 5. We consider truncated exponential distributions taking values in time interval $[0 - 10/rate]$, and then discretize them. The discrete input distributions are constructed in this manner.

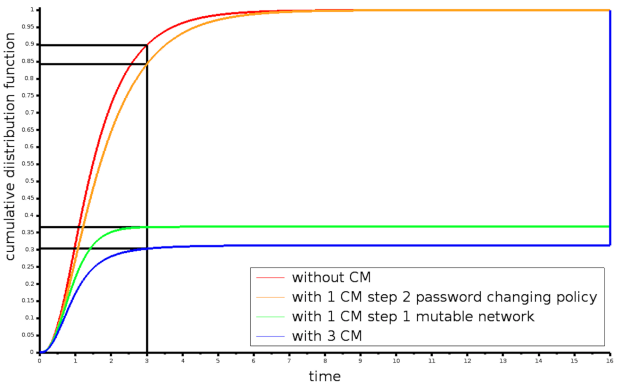

Fig. 3. Countermeasures effects

First we describe the behavior of the system according to the number of countermeasures applied. This corresponds to sub trees of the tree drawn on Fig. 2 with less *SI-AND* gates. On Fig. 3, one can see how the countermeasures modify the system behaviour during time interval $[0 - 16[$. Thus time 16 represents the infinity, and the events happening at time 16 indeed do not occur. The red curve shows the system behaviour when no countermeasure is applied. It corresponds to the attack tree with no *SI-AND* gate. At time $t = 3$ time unit, the probability that the attack succeeds is 0.89. If we consider the system, with only the *password changing policy* during the second step of the attack. The tree has now a single *SI-AND* gate. The computation results are shown on the

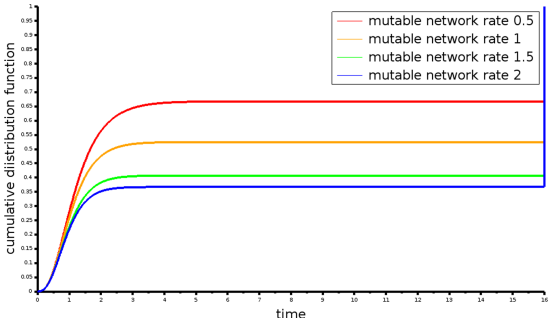

Fig. 4. Countermeasure distribution influence

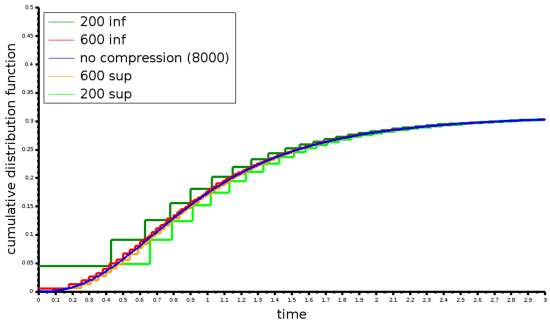

Fig. 5. Compression

orange curve. The probability at $t = 3$ time unit diminishes slightly to 0.84. On the green curve for the tree with only one countermeasure *mutable network*, the probability is only 0.37. This countermeasure is very efficient. The blue curve represents the system behaviour of the whole tree with three countermeasures. The probability at $t = 3$ time unit that the attack succeed is then only 0.30. Notice that for the green and the blue curves the cumulative distributions sum up to 1 at time 16 (∞). Therefore the probability that an attack occurs in the green curve is 0.37 while it is 0.30 in the blue curve. There is no jump at time 16 for the orange curve since the considered countermeasure is an input of an OR gate. Thus, the effect of the countermeasure is neutralized when the target exploitation happens in the other input.

Secondly we study the influence of a countermeasure distribution. We consider the tree with only one countermeasure *mutable network*. The other countermeasures are deleted (the operational time is defined as ∞). The Fig. 4 shows that the probability p that the attack succeeds increases when the rate λ of the countermeasure exponential distribution decreases. For $\lambda = 0.5$, $p = 0.67$, for $\lambda = 1$, $p = 0.52$, for $\lambda = 1.5$, $p = 0.41$ and for $\lambda = 2$, $p = 0.37$. Notice that the discrete input distributions are constructed as explained above.

Finally, we illustrate how compression influences the calculations on the full tree of Fig. 2. We apply the naive approach with linear complexity to construct bounding distributions. In the SEQ gate computations, the size of the results are bounded with different values. In each case a lower bound and an upper bound of the cumulative function are given. All the input distributions are discretized with 100 bins. If we do not bound the size of the results, the final result has about 8000 values. The results are shown in Fig. 5. The blue curve represents the cumulative function without any compression. We see that if we bound the result size to 600, the lower bound given by the red curve and the upper bound given by the orange curve still give usable values even for the small time values. The computations are less accurate when we bound the result size to 200. That is illustrated by the green curves.

4 Conclusion

In this paper, we extend the framework to analyze Attack Trees given in [11] to Attack Defense Trees with countermeasures. The completion times of attacks and countermeasures are defined by finite discrete random variables. The output distributions of the gates and finally the root of the tree are computed by a bottom-up approach. However the sizes of the output distributions can become quickly very large. We propose to derive bounding distributions with reduced sizes by means of the stochastic comparison method. In practise, we obtain fairly accurate bounds by compressing the output distribution even with linear computational complexity algorithms. Therefore the quantitative evaluation of Attack Trees with the presence of countermeasure can be efficiently provided by the proposed methodology.

We consider to extend this approach to the other extensions of Attack Trees and to the case when the basic events are not mutually independent.

References

1. Aït-Salaht, F., Castel-Taleb, H., Fourneau, J.-M., Pekergin, N.: Stochastic bounds and histograms for network performance analysis. In: Balsamo, M.S., Knottenbelt, W.J., Marin, A. (eds.) EPEW 2013. LNCS, vol. 8168, pp. 13–27. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40725-3_3
2. Arnold, F., Hermanns, H., Pulungan, R., Stoelinga, M.: Time-dependent analysis of attacks. In: Abadi, M., Kremer, S. (eds.) POST 2014. LNCS, vol. 8414, pp. 285–305. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54792-8_16
3. Aslanyan, Z., Nielson, F., Parker, D.: Quantitative verification and synthesis of attack-defence scenarios. In: IEEE CSF 2016, pp. 105–119 (2016)
4. Fourneau, J.M., Pekergin, N.: A numerical analysis of dynamic fault trees based on stochastic bounds. In: Campos, J., Haverkort, B.R. (eds.) QEST 2015. LNCS, vol. 9259, pp. 176–191. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22264-6_12

5. Gadyatskaya, O., Hansen, R.R., Larsen, K.G., Legay, A., Olesen, M.C., Poulsen, D.B.: Modelling attack-defense trees using timed automata. In: Fränzle, M., Markey, N. (eds.) FORMATS 2016. LNCS, vol. 9884, pp. 35–50. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44878-7_3
6. Jhawar, R., Lounis, K., Mauw, S.: A stochastic framework for quantitative analysis of attack-defense trees. In: Barthe, G., Markatos, E., Samarati, P. (eds.) STM 2016. LNCS, vol. 9871, pp. 138–153. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46598-2_10
7. Kordy, B., Mauw, S., Radomirovic, S., Schweitzer, P.: Attack-defense trees. J. Log. Comput. **24**(1), 55–87 (2014)
8. Kordy, B., Pièetre-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. Kumar, R., Ruijters, E., Stoelinga, M.: Quantitative attack tree analysis via priced timed automata. In: Sankaranarayanan, S., Vicario, E. (eds.) FORMATS 2015. LNCS, vol. 9268, pp. 156–171. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22975-1_11
10. Kumar, R., Stoelinga, M.: Quantitative security and safety analysis with attack-fault trees. In: HASE 2017, pp. 25–32 (2017)
11. Pekergin, N., Tan, S., Fourneau, J.-M.: Quantitative attack tree analysis: stochastic bounds and numerical analysis. In: Kordy, B., Ekstedt, M., Kim, D.S. (eds.) GraMSec 2016. LNCS, vol. 9987, pp. 119–133. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46263-9_8
12. Petrucci, L., Knapik, M., Penczek, W., Sidoruk, T.: Squeezing state spaces of (attack-defence) trees. In: Pang, J., Sun, J. (eds.) ICECCS 2019, pp. 71–80 (2019)
13. Roy, A., Seong, D., Kim, K.T.: Act:towards unifying the constructs of attack and defense trees. Securtiy Commun. Netw. **3**, 1–15 (2011)
14. Ruijters, E., Stoelinga, M.: Fault tree analysis: a survey of the state-of-the-art in modeling, analysis and tools. Comput. Sci. Rev. **15**, 29–62 (2015)
15. Schneier, B.: Attack trees: modeling security threats. Dr. Dobb's J. Softw. Tools **24**(12), 21–29 (1999)
16. Tancrez, J.S., Semal, P., Chevalier, P.: Histogram based bounds and approximations for production lines. Eur. J. of Oper. Res. **197**(3), 1133–1141 (2009)
17. Widel, W., Audinot, M., Fila, B., Pinchinat, S.: Beyond 2014: formal methods for attack tree-based security modeling. ACM Comput. Surv. **52**(4), 75:1–75:36 (2019)