TELECOM
ParisTech

Attack and defense dynamic modeling with BDMP – Extended version

Modélisation dynamique d'attaques et de défenses avec les BDMP – Version longue

Ludovic Piètre-Cambacédès
Marc Bouissou

2010D021

Septembre 2010

Attack and Defense Dynamic Modeling with BDMP - Extended Version

(Modélisation dynamique d'attaques et de défenses avec les BDMP - Version longue)

Ludovic Piètre-Cambacédès[a,b], Marc Bouissou[a,c]

[a]Électricité de France (EDF) R&D, 1, Avenue du Général de Gaulle, 92141 Clamart, France
[b]Institut Télécom, Télécom ParisTech, CNRS LTCI UMR 5141, 23 Avenue d'Italie CS51327, 75214 Paris Cedex 13, France
[c]École Centrale Paris, Grande Voie des Vignes, 92295 Châtenay-Malabry, France

September 14, 2010

Abstract

The BDMP (Boolean logic Driven Markov Processes) modeling formalism has recently been adapted from reliability engineering to security modeling. It constitutes an attractive trade-off in terms of readability, modeling power, scalability and quantification capabilities. This report develops and completes the theoretical foundations of such an adaptation and presents new developments on defensive aspects. In particular, detection and reaction modeling are fully integrated in an augmented theoretical framework. Different use-cases and quantification examples illustrate the relevance of the overall approach.

Résumé

Le formalisme de modélisation BDMP (*Boolean logic Driven Markov Processes*) initialement créé pour la modélisation de systèmes dynamiques dans le cadre d'études de fiabilité et de disponibilité a été adapté récemment au domaine de la sécurité. Ce formalisme constitue un compromis attractif en termes de lisibilité, de puissance de modélisation et de capacité à spécifier et quantifier des modèles de grande taille, adaptés à des systèmes réels. Ce rapport détaille la définition formelle des BDMP après adaptation au domaine de la sécurité. Il en rappelle les avantages pour la modélisation de différentes stratégies d'attaque (en séquence, en parallèle, avec des phases dédiées à tel ou tel type d'attaque) puis il présente des nouveaux développements sur les aspects défensifs. En particulier, détection de l'attaque et réactions du défenseur sont intégrées dans un cadre théorique enrichi. Différents exemples d'application illustrent la pertinence de cette approche sur le plan de la modélisation et des résultats de quantification obtenus. Enfin, même si le cadre ainsi présenté est déjà opérationnel, quelques pistes sont données pour gagner encore en réalisme et en rapidité des études de sécurité.

1   Introduction

Graphical attack formalisms are commonly used in security analysis to share standpoints between analysts, enhance their coverage in terms of scenarios, and help ordering them and the related system vulnerabilities by various quantifications. The authors have recently introduced a new approach based on BDMP (Boolean logic Driven Markov Processes) [1], adapting this formalism used in reliability engineering to attack modeling [2]. BDMP have proven to be an original and advantageous trade-off between readability, modeling power, scalability and quantification capabilities in their original domain [3]. The same advantages are expected from their adaptation to the security area. In this paper, we consolidate the theoretical foundations of such an adaptation, and extend it to take into account detection and reaction aspects in an integrated approach. Section 2 presents a brief state of the art in graphical attack modeling. Section 3 develops, on a theoretical and practical point of view, how BDMP can be changed to model attack scenarios. Section 4 focuses on defensive aspects, presenting the extension developed for detection and reaction modeling. Section 5 presents on-going and future work related to this new approach.

2   State of the Art

The clear interest of the computer security community for graphical attack modeling techniques has led to numerous proposals; they can be grouped into two categories, each being dominated by a specific model:

- *Static models*: also called structural models, they provide a global view of the attack, without being able to capture its evolution in time. The dominant type of model is the Boolean-logical tree based approach. Generally known as attack trees [4, 5], they are present in the literature under different variations: threat trees [6], vulnerability trees [7], etc.

- *Dynamic models*: also called behavioral models, they take into account dependance aspects such as sequences or reactions. Richer than static models, they can be built by hand only in very simple cases. There are two approaches in the other cases:

- The first one is based on detailed state-graphs capturing the possible evolutions of an attack, automatically generated from formal specifications. Such approaches, initiated by Sheyner *et al.* with attack graphs [8] and followed by other relevant approaches (e.g., [9, 10]), are not graphical models *per se* as they are not directly designed to be graphically manipulated by analysts.
- The second relies on compact and high-level graphical formalisms, designed to efficiently represent dynamic aspects like sequences or reactions, and to be directly usable by human analysts. In this category, Petri net-based approaches are the most widely known. Attack nets, described ten years ago by McDermott [11], or PE nets, a more recent approach with a complete software support [12], are two good representatives.

Each approach allows for a different balance in terms of modeling power, readability, scalability and quantification capabilities. Static models are usually very readable but are lacking in their modeling power and quantification capabilities. Dynamics models are more interesting for these aspects, but often have their own limits in terms of clarity and scalability. Note that these statements are also relevant in the domain of reliability and safety modeling [13, 14], where similar approaches have been historically first used, modeling system component failures instead of attacker actions and security events.

3 The BDMP Formalism Applied to Attack Modeling

3.1 Foundations

Originally, BDMP are a formalism which combines the readability of classical fault trees with the modeling power of Markov chains [1]. Generally speaking, it changes the fault tree semantics by augmenting it with a special kind of links called *triggers*, and associating its leaves to Markov processes, dynamically selected in function of the states of some other leaves. This allows for sequences and simple dependencies modeling, while enabling efficient quantifications. The original definition, the mathematical properties and different examples are provided in [1]. In this section, we present the main elements of theory and features offered by a straightforward adaptation of BDMP to security modeling, summing up and completing ref. [2].

3.1.1 The components of BDMP

Informally, "triggered" Markov processes (noted P_i and presented in this section) are associated to the leaves i of an attack tree $\mathcal{A}$. Each process has two modes: *Idle* and *Active* (formally noted 0 and 1). The former models an on-going event, in general an attacker action, the latter is used when nothing is in progress. The mode of a given P_i is a Boolean function of the states of the other processes. Fig. 1 represents a simple exemple of a BDMP, with its typical graphical components.
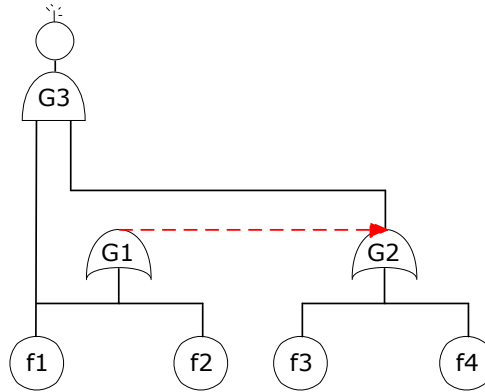

Figure 1: A small BDMP

More formally, a security-oriented BDMP is a set $\{\mathcal{A}, r, T, P\}$ composed of:

- an attack tree $\mathcal{A} = \{E, L, g\}$, where:

  - $E = G \cup B$, with G a set of logical gates, and B a set of basic security events (e.g. attacker actions), corresponding to the leaves of the BDMP,
  - $L \subset G \times E$ is a set of oriented edges, such that (E, L) is a directed acyclic graph with $\forall i \in G, sons(i) \neq \emptyset$ and $\forall j \in B, sons(j) = \emptyset$, with $E \xrightarrow{sons} 2^E, sons(i) = \{j \in E / (i, j) \in L\}$,
  - $g : G \to \mathbb{N}^*$ is a function defining the parameter k of the gates which are all considered to be k/n logical gates ($k = 1$ for OR gates, $k = n$ for AND gates, with n the number of sons);

- r, the final attacker's objective. Formally, it corresponds to a top of (E, L);

- a set of triggers $T \subset (E - \{r\}) \times (E - \{r\})$ such that $\forall (i, j) \in T, i \neq j$ and $\forall (i, j) \in T, \forall (k, l) \in T, i \neq k \Rightarrow j \neq l$. If i is called origin and j target, it means that origin and target of a trigger must differ, and that two triggers cannot have the same target. Triggers are represented by dotted arrows;

- a set P of triggered Markov processes $\{P_i\}_{i \in B}$. Each P_i is defined as a set $\left\{ Z_0^i(t), Z_1^i(t), f_{0 \to 1}^i, f_{1 \to 0}^i \right\}$ where:

  - $Z_0^i(t)$ and $Z_1^i(t)$ are two homogeneous Markov processes with discrete state spaces. For k in $\{0, 1\}$, the state space of $Z_k^i(t)$ is $A_k^i(t)$. Each $A_k^i(t)$ contains a subset $S_k^i(t)$ which corresponds to success or realization states of the basic security event modeled by the process P_i,
  - $f_{0 \to 1}^i$ and $f_{1 \to 0}^i$ are two "probability transfer functions" defined as follows:
    * for any $x \in A_0^i$, $f_{0 \to 1}^i(x)$ is a probability distribution on A_1^i such that if $x \in S_0^i$, then $\sum_{j \in S_1^i} (f_{0 \to 1}^i(x))(j) = 1$,
    * for any $x \in A_1^i$, $f_{1 \to 0}^i(x)$ is a probability distribution on A_0^i such that if $x \in S_1^i$, then $\sum_{j \in S_0^i} (f_{1 \to 0}^i(x))(j) = 1$.

Triggers and P_is are intimately linked, as the P_is switch instantaneously between modes, via the relevant probability transfer function, according to the state of some externally defined Boolean variables, called process selectors (defined in the next paragraph). The process selectors are defined by means of triggers. In the simple cases where only one trigger is present in the model, a trigger modifies the mode of the P_is associated to the leaves of the sub-tree it points at when its origin changes from false to true: the modes are then switched from *Idle* to *Active*. When several triggers are present, their effects are combined following the formal relations given in the next section. These mechanisms model the progress of the attacker in the attack scenarios captured by the overall BDMP.

3.1.2 The three families of Boolean functions of the time

A BDMP defines a global stochastic process, modeling the evolution of an attack and the dynamic behavior of its perpetrator. Each element i of $\mathcal{A}$ is associated to three Boolean functions of time: a structure function $S_i(t)$, a process selector $X_i(t)$ and a relevance indicator $Y_i(t)$. The three families of these functions are defined as follows (note that to simplify reading, the time t is not indicated but should appear everywhere):

- $(S_i)_{i \in E}$ is the family of the structure functions. They respect the following relation:

$$\forall i \in G, S_i \equiv (\sum_{j \in sons(i)} S_j \geq g(i)) \text{ and } \forall j \in B, S_j \equiv (Z_{X_j}^j \in S_{X_j}^j)$$

  with X_j indicating the mode in which P_j is at time t. $S_j = 1$ corresponds to the realization of a basic security event (like an attacker action success);

3

- $(X_i)_{i \in E}$ are the mode selectors, indicating which mode is chosen for each process. If i is a top of $\mathcal{A}$, then $X_i = 1$ else:

$$X_i \equiv \neg\left[(\forall x \in E, (x,i) \in L \Rightarrow X_x = 0) \vee (\exists x \in E/(x,i) \in T \wedge S_x = 0)\right].$$

This means that $X_i = 1$ except if the origin of a trigger pointing at i has its structure function equal to 0, or if i has at least one parent and all its parents have their process selector equal to 0;

- $(Y_i)_{i \in E}$ are the relevance indicators. They are used to mark the processes to be "trimmed" during the processing of the Markov chain when exploring the possible sequences. Trimming strongly reduces the combinatorial explosion while yielding exact results in our assumptions (cf. the next paragraph and Section 3.4). If $i = r$ (final objective), then $Y_i = 1$, else:

$$Y_i \equiv (\exists x \in E/(x,i) \in L \wedge Y_x = 1 \wedge S_x = 0) \vee (\exists y \in E/(i,y) \in T \wedge S_y = 0).$$

This formally says that $Y_i = 1$ if and only if:

- $i = r$,
- or i has at least one "relevant parent" whose $S_i = 0$,
- or i is the origin of at least one trigger pointing at an element whose $S_i = 0$.

3.1.3   Mathematical properties

A BDMP can be seen as a robust mathematical formalism thanks to the two following theorems:

Theorem 1. *The functions $(Y_i), (X_i), (Y_i)$ are computable for all $i \in E$ whatever the BDMP structure.*

Theorem 2. *Any BDMP structure associated to an initial state defined by the modes and the P_i states, uniquely defines a homogeneous Markov process.*

The proof for these theorems can be found in [1]. In addition to their robustness, BDMP allow for a dramatic combinatory reduction by relevant event filtering, thanks to the trimming mechanism associated to the (Y_i) values. This mechanism can be illustrated as follows: in Fig. 2, after a basic security event P_i is realized, all the other $P_{j \neq i}$ are no longer relevant: nothing is changed for r if we inhibit them. The number of sequences leading to the top objective is n if the relevant events are filtered $((P_1, Q), (P_2, Q), ...)$; it is exponential otherwise $((P_1, Q), (P_1, P_2, Q), (P_1, P_3, Q), ...)$.
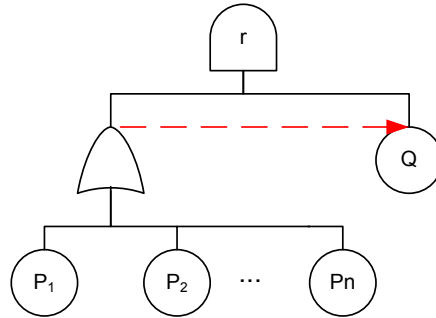

Figure 2: A BDMP for which relevant event filtering is particularly efficient

Theorem 3. *If the (P_i) are such that $\forall i \in B, \forall t, \forall t' \geq t, S_i(t) = 1 \Rightarrow S_i(t') = 1$ (which is always true in our case), then $Pr(S_r(t) = 1)$ is unchanged whether irrelevant events (with $Y_i = 0$) are trimmed or not.*

The proof of this last theorem is also given in [1]. It implies that trimming on the basis of the (Y_i) does not change the quantitative values of interest (cf. Section 3.4). Moreover, it corresponds to the natural and rational behavior of the attacker.

3.1.4 The basic leaves and their triggered Markov processes

The definition of three kinds of leaves is sufficient to offer large attack modeling capabilities. Their triggered Markov processes are represented informally in Tab. 1.

- The "Attacker Action" (AA) leaf models an attacker step towards the accomplishment of his objective. The *Idle* mode means that the action has not at this stage been tried by the attacker. The *Active* mode corresponds to actual attempts for which the time needed to succeed is exponentially distributed with a parameter λ. When (X_i) changes from 0 (*Idle*) to 1 (*Active*), the leaf state goes from *Potential* to *On-going*; when (X_i) goes back from 1 to 0, if the attack has not succeeded, the leaf state goes back to *Potential*, if it has succeeded, the leaf comes back to the *Success* state of the *Idle* mode. Formally, the probability transfer functions are:

$$\begin{aligned}
f_{0\to1}(P) &= \{\Pr(O) = 1, \Pr(S) = 0\}, \\
f_{0\to1}(S) &= \{\Pr(O) = 0, \Pr(S) = 1\}, \\
f_{1\to0}(O) &= \{\Pr(P) = 1, \Pr(S) = 0\}, \\
f_{1\to0}(S) &= \{\Pr(P) = 0, \Pr(S) = 1\}.
\end{aligned}$$
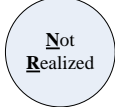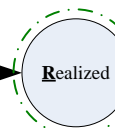
- The "Instantaneous Security Event" (ISE) leaf models a basic security event that can happen instantaneously with a probability γ, when the leaf switches from the *Idle* mode to the *Active* mode. In the *Idle* mode, the event cannot occur and the leaf stays in the state *Potential*. In the *Active* mode, the event is either *Realized* or *Not Realized*. State changes are necessarily the result of changes in (X_i). Formally, the probability transfer functions are:

$$\begin{aligned}
f_{0\to1}(P) &= \{\Pr(NR) = 1 - \gamma, \Pr(R) = \gamma\}, \\
f_{0\to1}(R) &= \{\Pr(NR) = 0, \Pr(R) = 1\}, \\
f_{1\to0}(R) &= \{\Pr(NR) = 0, \Pr(R) = 1\}, \\
f_{1\to0}(NR) &= \{\Pr(P) = 1, \Pr(R) = 0\}.
\end{aligned}$$

- The "Timed Security Event" (TSE) leaf models a timed basic security event the realization of which impacts the attacker's progress, but which is not under the attacker's direct control. The time needed for its realization is exponentially distributed. When the leaf comes back to the *Idle* mode, the leaf state can then be either *Realized* or *Not Realized*, depending on whether the TSE has occurred or not in *Active* mode. If unrealized, it is up to the analyst to decide if a realization is then possible in *Idle* mode, by using a $\lambda' \neq 0$. This can be useful when using phased approaches as described in Section 3.3. Formally, the transfer functions are as follows:

$$\begin{aligned}
f_{0\to1}(P) &= \{\Pr(NR) = 1, \Pr(R) = 0\}, \\
f_{0\to1}(NR) &= \{\Pr(NR) = 1, \Pr(R) = 0\}, \\
f_{0\to1}(R) &= \{\Pr(NR) = 0, \Pr(R) = 1\}, \\
f_{1\to0}(NR) &= \{\Pr(NR) = 1, \Pr(R) = 0\}, \\
f_{1\to0}(R) &= \{\Pr(NR) = 0, \Pr(R) = 1\}.
\end{aligned}$$

Table 1: The three basic security leaves for attack modeling

Leaf type & icon	*Idle* Mode ($X_i=0$)	Transfer between modes	*Active* Mode ($X_i=1$)
Attacker Action	Potential Success	P⇔O (with $Pr = 1$) S⇔S (with $Pr = 1$)	On-going $\xrightarrow{\lambda}$ Success, $S_i \leftarrow 1$
Instantaneous Security Event (ISE!)	Potential Realized	P⇒NR (with $Pr=1-\gamma$) P⇒R (with $Pr = \gamma$) R⇔R (with $Pr = 1$) P⇐NR (with $Pr = 1$)	Not Realized $\xrightarrow{\lambda}$ Realized, $S_i \leftarrow 1$
Timed Security Event (TSE)	Potential Not Realized $\xrightarrow{\lambda'}$ Realized, $S_i \leftarrow 1$	P⇒NR (with $Pr = 1$) NR⇔NR (with $Pr=1$) R⇔R (with $Pr = 1$)	Not Realized $\xrightarrow{\lambda}$ Realized, $S_i \leftarrow 1$

3.2 Sequence Modeling

The triggers allow for an efficient and readable modeling of the sequential nature of attacks: often, some actions or events need to be undertaken or realized first before further steps in the attack process can be attempted. Fig. 3 presents a simple example with a sequence of three actions with such a constraint, based on an Operating System (OS) attack. Reference [2] proposes an alternative example, modeling the attack of a Remote Access Server (RAS), while a complete use-case is presented in Section 3.4.
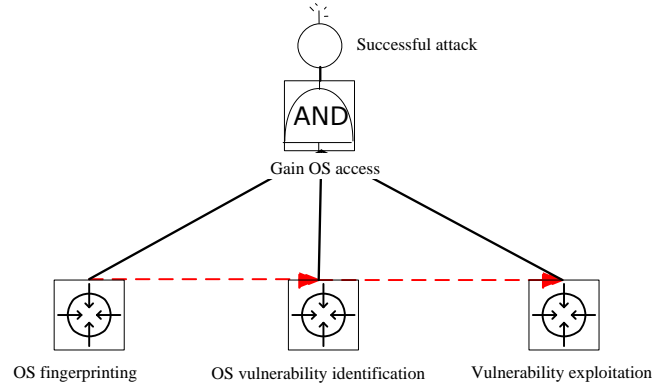
Figure 3: A simple OS attack

3.3 Modeling of Concurrent or Exclusive Alternatives

For a given intermediate objective, an attacker may have different alternatives. A natural way of modeling this with BDMP and classical attack trees is with OR gates. Fig. 4 represents two different approaches with an example dealing with OS fingerprinting. On the left side, a simple OR gate is used: passive and active techniques are tried simultaneously, which may not reflect a realistic attacker behavior. Passive techniques, being more discrete, would normally be tried first and, if not successful, given up after some time for active ones. Triggers cannot model such a behavior. "Phase leaves", used on the right side of Fig. 4, allow this behavior to be modeled; their formal definition is given in [2].
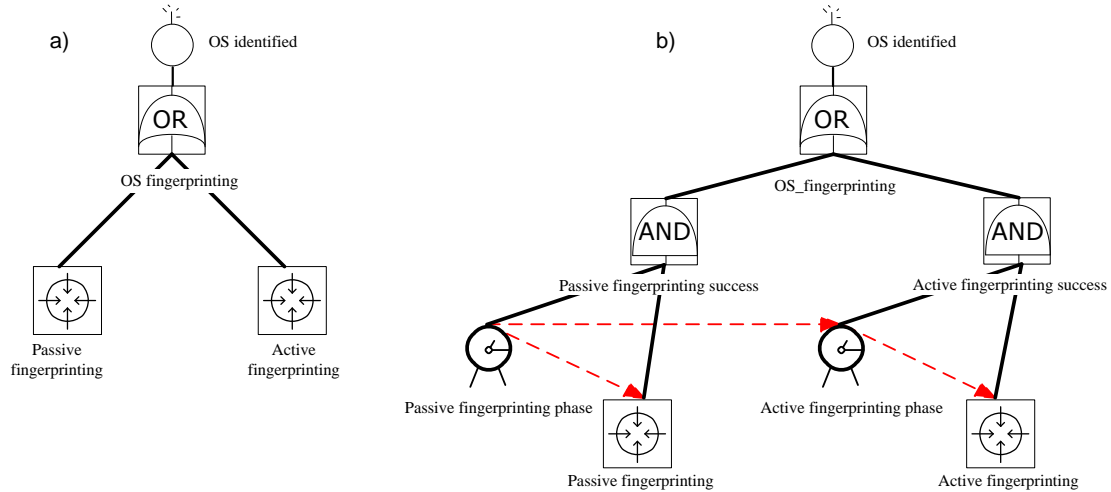
Figure 4: Sequence of a simplified OS attack

7

3.4 Diverse and Efficient Quantifications: Principles and Use-case

The interest of BDMP does not only lie in the possibility to represent sequences. They enable diverse time-domain quantifications, including the probability for an attacker to reach his objective in a given time or the overall mean time for the attack to succeed. In addition, BDMP analysis yields the enumeration of all the possible attack paths, ordered by their probability of occurrence in a given time. Such results can be efficiently computed thanks to an original analytical method developed for large Markov models, and thus applicable to BDMP [15]. Indeed, as explained previously, BDMP are high-level representations of potentially large Markov chains; however, the treatment of such chains is usually confronted with state-space explosion. It is overcome using a path-based approach, exploring the sequences leading to the undesirable states. Such an approach enables exact calculations for small models by exhaustive exploration. For larger models, it is possible to obtain controlled approximations by limiting the sequence exploration to those having a probability greater than a given threshold. In both cases, the probability of the explored sequences is computed by the closed form expression given in [16]. Sequence exploration takes advantage of the trimming mechanism described in Section 3.1, which leads to a strong combinatorial reduction.

More concretely, the analyst must define the λ parameters of the exponential distributions and the γ parameters of the ISE leaves. Defining the λs is done by reasoning in terms of Mean Time To Success (MTTS), i.e. $1/\lambda$, like in [17, 18, 19]. The γs are also set subjectively. The parameters should be estimated based on the intrinsic difficulty of the attacker actions, his estimated skills and resources, and the level of system protection. We have used the KB3 workbench [3] for the model construction and quantitative treatments in this report. Fig. 5 models the attack of a password-protected file, of which a copy has been stolen. In our scenario, obtaining the password is the only way to access its content, needed by the attacker within a week (this may take place in a call for tender in a competitive environment). The parameters chosen are indicated in Tab. 2.

Table 2: Parameters of the use-case

Leaf name	Type	Parameter	Remark
Guessing, Dictionary	AA	$\lambda = 0\ s^{-1}$	Considered as impossible (long and random)
Bruteforce	AA	$\lambda = 3.802 \times 10^{-7}\ s^{-1}$	MTTS $(1/\lambda) \approx$ a month
Social_eng_phase	Phase	172,800 s.	Mean duration = 2 days
Generic_reconnaissance	AA	$\lambda = 1.157 \times 10^{-5}\ s^{-1}$	MTTS $(1/\lambda) \approx$ 1 day
Email_trap_execution	AA	$\lambda = 1.157 \times 10^{-5}\ s^{-1}$	MTTS $(1/\lambda) \approx$ 1 day (regular nomad access)
Phone_trap_execution	AA	$\lambda = 5.787 \times 10^{-6}\ s^{-1}$	MTTS $(1/\lambda) \approx$ 2 days
User_trapped	ISE	$\gamma = 0.33$	1 out of 3 (targeted attack but cautious user)
Keylogger_phase	Phase	432,000 s.	Mean duration = 5 days
Remote_phase	Phase	172,800 s.	Mean duration = 2 days
Payload_crafting	AA	$\lambda = 5.787 \times 10^{-6}\ s^{-1}$	MTTS $(1/\lambda) \approx$ 2 days
Crafted_attachement_open	TSE	$\lambda = 1.157 \times 10^{-5}\ s^{-1}$	MTTS $(1/\lambda) \approx$ 1 day
Appropriate_payload	ISE	$\gamma = 0.1$	1 out of 10 (still many unknown factors)
Physical_phase	Phase	259,200 s.	Mean duration = 3 days
Physical_reconnaissance	AA	$\lambda = 5.787 \times 10^{-6}\ s^{-1}$	MTTS $(1/\lambda) \approx$ 2 days
Keylogger_local_installation	AA	$\lambda = 1.157 \times 10^{-5}\ s^{-1}$	MTTS $(1/\lambda) \approx$ 1 day
Password_intercepted	TSE	$\lambda = 1.157 \times 10^{-5}\ s^{-1}$	MTTS $(1/\lambda) \approx$ 1 day

Such parameters lead to a probability of success in a week of 0.422, with an overall MTTS of 22 days. An exhaustive exploration gives 654 possible sequences; Tab. 3 shows a representative excerpt. The beginning of a phase is marked as "<phase>" and its end as "</phase>". Even if phases are not basic security events, they are fully part of the sequences as they structure their chronology. The same applies to the leaves that are realized unnecessarily; they are marked in *italics*. As one can see, most of the sequences include one or more unnecessary actions or events that have no effect on the global success of the attack and as such, these sequences are non-minimal. The minimal sequences are called success sub-sequences, or SSS. Seq. 1 to 4 are minimal and weigh probabilistically 47% of all the sequences. Seq. 5 and 6 are good examples of non-minimal sequences. Bruteforce is a specific leaf as it is also the only single element SSS. It
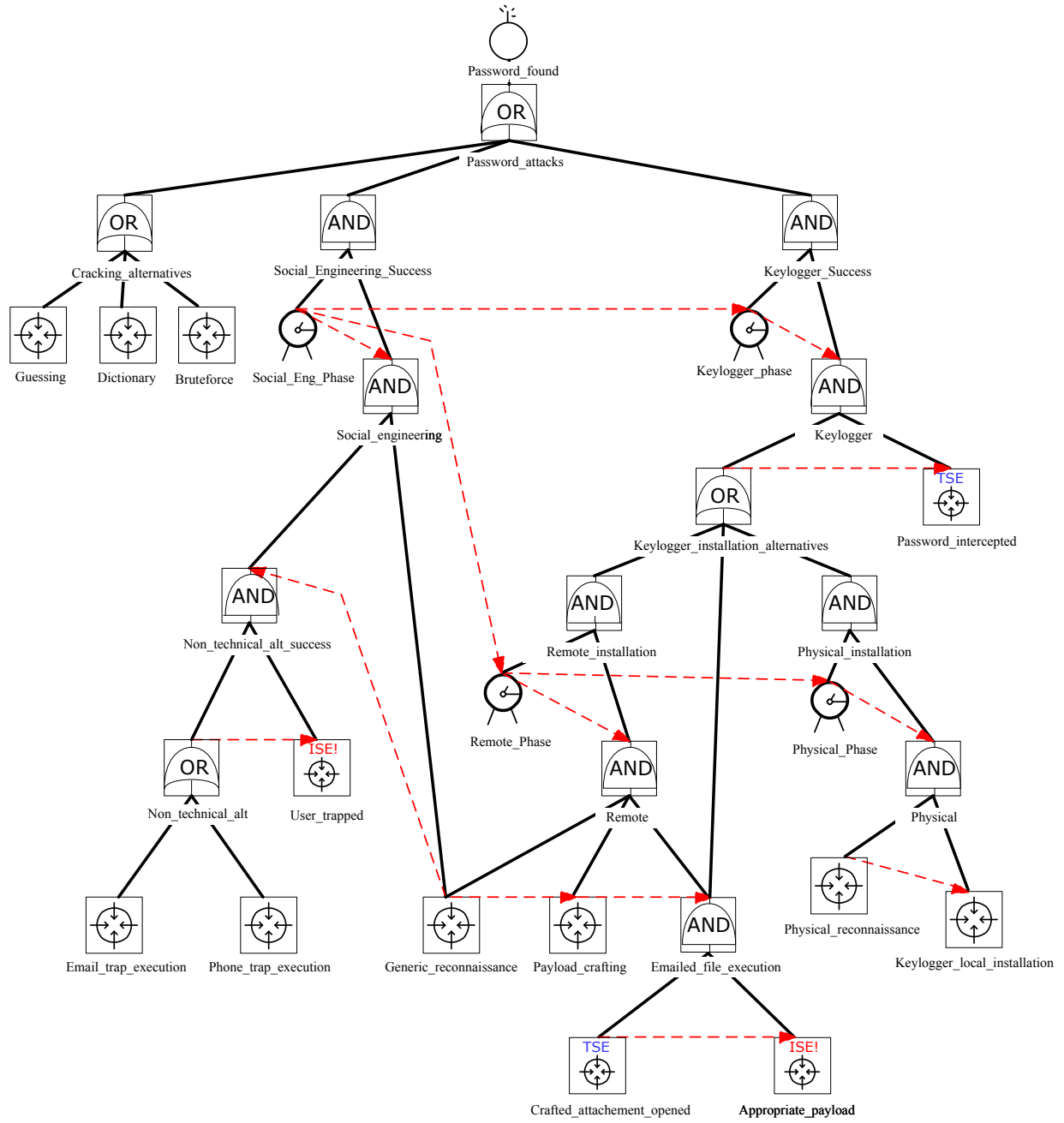
Figure 5: Attack of a password-protected file

Table 3: Selection of sequences with quantifications

	Sequences	Probability in a week	Average duration (s)	Contrib.
1	<Social Eng>Generic reconn., Email trap exec., User trapped	1.059×10^{-1}	9.889×10^4	25.1%
2	<Social Eng>Generic reconn., Phone trap exec., User trapped	5.295×10^{-2}	9.889×10^4	12.5%
3	Bruteforce	2.144×10^{-2}	5.638×10^4	5.1%
4	<Social Eng></Social Eng><Keylogger><Remote></Remote><Physical> Physical reconn., Keylogger local installation, Password intercepted	1.749×10^{-2}	2.976×10^5	4.1%
5	<Social Eng></Social Eng><Keylogger> <Remote>*Generic reconnaissance* </Remote><Physical>Physical reconnaissance, Keylogger local installation, Password intercepted	1.350×10^{-2}	3.677×10^5	3.2%
6	<Social Eng>*Generic reconnaissance, Email trap execution, User trapped(failure)*, Bruteforce	1.259×10^{-2}	2.610×10^5	3.0%
...				
20	<Social Eng></Social Eng><Keylogger><Remote>Generic reconnaissance, Payload crafting, Appropriate payload, Password intercepted	2.500×10^{-3}	2.761×10^5	0.6%
...				
34	<Social Eng></Social Eng><Keylogger> <Remote>Generic reconn., *Payload crafting* </Remote> <Physical>*Crafted attachement opened, Appropriate payload*, Physical reconn., Keylogger local installation, Password intercepted	1.506×10^{-3}	4.594×10^5	0.4%

appears directly as a minimal sequence in line 3, but also ends numerous non-minimal sequences. In fact, the consolidated contribution of all the sequences ended by bruteforce weighs 40% of all the sequences. Such a strong weight despite bruteforce's large MTTS is due to the absence of other steps to be fulfilled. This points to a more generic statement: a complete analysis should not only use the list of sequences, but also consider complementary views, including the consolidated contributions of the SSS. Seq. 3 to 19 involve only two SSS; seq. 20 relies on a new SSS, then one has to wait until seq. 34 to find another one. This latter sequence illustrates the specificity of TSE leaves, which are able to be realized in Idle mode if the leaf has been Active at least once.

3.5 Hierarchical and scalable analysis

It is possible to choose for each attacker action the depth of analysis, leading to different breakdowns depending on the analysis needs. This hierarchical behavior is a powerful property directly inherited from the attack tree formalism. In Fig. 5, the password cracking alternatives have been broken down quite roughly into three techniques which might have been decomposed themselves into much finer possibilities; on the other hand, the social engineering and the keylogger sub-trees are slightly more developed. More detailed breakdowns would have been possible. In fact, BDMP with more than 100 leaves are routinely processed in reliability studies [3]: the method is also scalable for security applications.

4 Integrating Defensive Aspects: Detection and Reaction

Holistic approaches to security generally cover protection, detection and reaction. The level of protection can be considered as intrinsically reflected by the BDMP structure, modeling only possible ways for attacks, and its leaves' parameters (λs and γs), reflecting the attack difficulty confronted with a given protection level. This section presents the specifically tailored extensions to BDMP needed to model detection and reaction aspects.

4.1 The IOFA detection decomposition

The integration of detection in a dynamic perspective has led us to distinguish four types of detection for the AA and TSE leaves, differentiated by the moment when the detection takes place. Type I (Initial)

detections take place at the very start of the attacker actions or of the events modeled; type O (On-going) take place during the attacker attempts or during the events modeled; type F (Final) detections take place at the moment the attacker succeeds in an action or when an event is realized; Type A (A posteriori) detections take place once an action or an event has been realized, based on the traces left by such an action or event.

Each of them has a specific relevance in a security context. Such distinction allows for a fine-tuned and complete modeling of detection; it is designated by the acronym IOFA. ISE leaves have been treated slightly differently with two distinct detections, depending on the realization outcome.

4.2 Extending the theoretical framework

In order to model detections and reactions, we extend the framework of Section 3.1 by:

- associating to each element a Boolean D_i, called Detection status indicator;

- replacing the *Active* mode by an *Active Undetected* mode and an *Active Detected* mode;

- selecting the mode on the basis of $X_i D_i$, and not only X_i, as described in Tab. 4 (note that in the formal notations of the following sections, 0 in subscript corresponds to the *Idle* mode and covers $X_i D_i = 00$ or 01);

- extending the leaves' triggered Markov processes with new states, transitions, and probability transfer functions, modeling detections and reactions.

Table 4: The new compound process selector $X_i D_i$ and the corresponding modes

$X_i D_i$	00	01	10	11
Mode	Idle		Active Undetected (AU)	Active Detected (AD)

4.2.1 Detection and reaction in the triggered Markov processes

In this framework, a P_i is a set $\left\{ Z_0^i(t), Z_{10}^i(t), Z_{11}^i(t), f_{0\to10}^i, f_{0\to11}^i, f_{10\to11}^i, f_{10\to0}^i, f_{11\to0}^i \right\}$ where:

- $Z_0^i(t), Z_{10}^i(t), Z_{11}^i(t)$ are three homogeneous Markov processes with discrete state spaces. For $k \in \{0, 10, 11\}$, the state space of $Z_k^i(t)$ is A_k^i. Each A_k^i contains a subset S_k^i which corresponds to success or realization states of the basic security event modeled by the process P_i, and a subset D_k^i which corresponds to detected states.

- $f_{0\to10}^i, f_{0\to11}^i, f_{10\to11}^i, f_{10\to0}^i, f_{11\to0}^i$ are five "probability transfer functions" defined as follows:

  - for any $x \in A_0^i$, $f_{0\to10}^i(x)$ is a probability distribution on A_{10}^i, such that if $x \in S_0^i$, then $\sum_{j\in S_{10}^i}(f_{0\to10}^i(x))(j) = 1$, and if $x \in D_0^i$, then $\sum_{j\in D_{10}^i}(f_{0\to10}^i(x))(j) = 1$;

  - for any $x \in A_0^i$, $f_{0\to11}^i(x)$ is a probability distribution on A_{11}^i, such that if $x \in S_0^i$, then $\sum_{j\in S_{11}^i}(f_{0\to11}^i(x))(j) = 1$, and if $x \in D_0^i$, then $\sum_{j\in D_{11}^i}(f_{0\to11}^i(x))(j) = 1$;

  - for any $x \in A_{10}^i$, $f_{10\to11}^i(x)$ is a probability distribution on A_{11}^i, such that if $x \in S_{10}^i$, then $\sum_{j\in S_{11}^i}(f_{10\to11}^i(x))(j) = 1$, and if $x \in D_{10}^i$, then $\sum_{j\in D_{11}^i}(f_{10\to11}^i(x))(j) = 1$;

  - for any $x \in A_{11}^i$, $f_{11\to0}^i(x)$ is a probability distribution on A_0^i, such that if $x \in S_{11}^i$ then $\sum_{j\in S_0^i}(f_{11\to0}^i(x))(j) = 1$, and if $x \in D_{11}^i$, then $\sum_{j\in D_0^i}(f_{11\to0}^i(x))(j) = 1$;

  - for any $x \in A_{10}^i$, $f_{10\to0}^i(x)$ is a probability distribution on A_0^i, such that if $x \in S_{10}^i$ then $\sum_{j\in S_0^i}(f_{10\to0}^i(x))(j) = 1$, and if $x \in D_{10}^i$, then $\sum_{j\in D_0^i}(f_{10\to0}^i(x))(j) = 1$.

Note that $f_{11\to10}^i$ is not defined: an attacker once detected cannot subsequently become undetected.

The triggered Markov processes of Section 3.1 are re-engineered to integrate detection and reaction features, as presented in Tab. 5, 6 and 7. They support the IOFA decomposition: detection is possible for a given attacker action or timed security event at its very start, during the attempts, at success time and even a posteriori. Transition parameters associated to detection are marked with a "D" in subscript. In the case of AA and TSE leaves, this letter is followed in parenthesis by the type of detection (i.e. I, O, F or A) they characterize; in the case of ISE leaves, it is followed by the characterized outcome ("/R" in case of realization, "/NR" in case of bad outcome for the attacker). The success and realization parameters are linked to the detection status of the leaf: "/D" in subscript means "having been detected" whereas "/ND" means "having not been detected". Discs with dotted circumferences represent "instantaneous" states whereas full discs are regular timed states. By instantaneous states we mean either:

- Artificial states introduced for the sake of clarity, but which could be removed by merging the incoming timed transitions with the outgoing instantaneous transitions into single timed transitions (e.g. the state SPD in Tab. 5),

- Special "triggering" states which have been introduced to change the D_i values, and to trigger mode changes based on internal leaves evolution. For instance in Tab. 5, in AU mode, an arrival either in the *Detected* or the *Success Detected* states triggers an instantaneous mode switch towards the AD mode: both arrivals set the Detection indicator status D_i at 1, passing the Boolean $X_i D_i$ value, used to select the mode, from 10 to 11. Such "triggering" instantaneous states are represented by striped discs.

4.2.2 Reaction "propagation"

The extended Markov model of the Attacker Action leaf in AU mode (cf. Tab. 5) is a good illustration on how detection is taken into account "within" a given leaf, and can provoke a local mode switch towards the AD mode. This changes the leaf parameter $\lambda_{S/ND}$ to a new value $\lambda_{S/D}$, turning the action more difficult or even impossible, if $\lambda_{S/D} = 0$, when the attacker is detected. The same applies for the other leaves. But such mode switches can also be provoked "externally", i.e. by a detection having occurred at the level of a different leaf. In fact, the following possibilities can be distinguished:

- the detection has a strictly local incidence: only the detected attacker action or security event is affected, the rest of the BDMP is unchanged, i.e. the other leaves keep the same parameters λs and γs;

- the detection has an extended incidence, changing not only the on-going detected leaf parameters but also a specific set of other leaves in the BDMP;

- the detection has a global incidence: in case of detection, all the D_i are set to 1, meaning that all the future attacker actions or security events will be in *Detected* mode, with the associated parameters.

This last option is the one that has been adopted in this paper: it is both meaningful in terms of security and straightforward in terms of formalization and implementation. Note that the intermediate option, especially relevant when dealing with multi-domain systems, has been explored by the authors and can be implemented by the introduction of "detection triggers". The associated developments will be presented in a separate publication.

4.2.3 Use-case taking into account detections and reactions

The use-case of Section 3.4 has been completed by adding detection possibilities and reactions for the leaves indicated in Tab. 8 with their corresponding parameters.

Globally, the introduction of detections and reactions reduces the probability of success within a week by about 14%, from 0.423 to 0.364. This modest reduction can be explained by the fact that the most probable success sequence, the single off-line bruteforce, is not subject to detection. In fact, even with systematic detections and perfect reactions (the attack is stopped), the attacker would still have a 0.201 probability

12

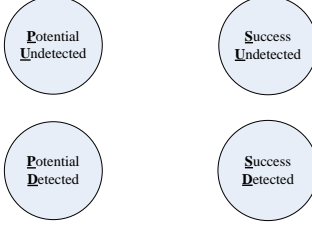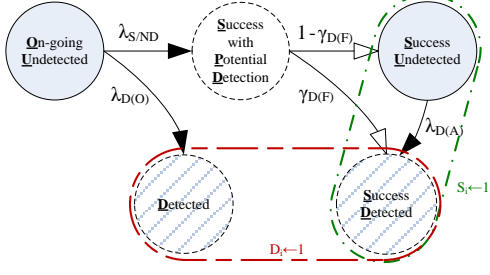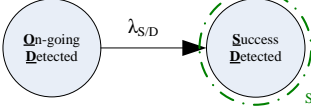Table 5: The triggered Markov processes of the Attacker Action (AA) leaves

Attacker Action (AA)	
Markov processes	**Probability transfer functions**
Idle $(Z_0^i(t))$ Potential Undetected Success Undetected Potential Detected Success Detected	$f_{0\to10}^i(PU)=\{Pr(OU)=1-\gamma_{D(I)},\ Pr(D)=\gamma_{D(I)},\ Pr(SD)=0,\ Pr(SU)=0\}$ $(PD)=\{Pr(OU)=0,\ Pr(D)=1,\ Pr(SD)=0,\ Pr(SU)=0\}$ $(SU)=\{Pr(OU)=0,\ Pr(D)=0,\ Pr(SD)=0,\ Pr(SU)=1\}$ $(SD)=\{Pr(OU)=0,\ Pr(D)=0,\ Pr(SD)=1,\ Pr(SU)=0\}$ $f_{0\to11}^i(PU)=\{Pr(OD)=1,\ Pr(SD)=0\}*$ $(PD)=\{Pr(OD)=1,\ Pr(SD)=0\}$ $(SU)=\{Pr(OD)=0,\ Pr(SD)=1\}*$ $(SD)=\{Pr(OD)=0,\ Pr(SD)=1\}$
Active Undetected $(Z_{10}^i(t))$ On-going Undetected $\xrightarrow{\lambda_{S/ND}}$ Success with Potential Detection $\xrightarrow{1-\gamma_{D(F)}}$ Success Undetected $\quad S_t\leftarrow1$ $\lambda_{D(O)}$ $\gamma_{D(F)}$ $\lambda_{D(A)}$ Detected Success Detected $\quad S_t\leftarrow1$ $D_t\leftarrow1$	$f_{10\to11}^i(OU)=\{Pr(OD)=1,\ Pr(SD)=0\}*$ $(D)=\{Pr(OD)=1,\ Pr(SD)=0\}**$ $(SD)=\{Pr(OD)=0,\ Pr(SD)=1\}**$ $(SU)=\{Pr(OD)=0,\ Pr(SD)=1\}*$ $f_{11\to0}^i(OD)=\{Pr(PU)=0,\ Pr(PD)=1,\ Pr(SD)=0,\ Pr(SU)=0\}$ $(SD)=\{Pr(PU)=0,\ Pr(PD)=0,\ Pr(SD)=1,\ Pr(SU)=0\}$ $f_{10\to0}^i(OU)=\{Pr(PU)=1,\ Pr(PD)=0,\ Pr(SD)=0,\ Pr(SU)=0\}$ $(SU)=\{Pr(PU)=0,\ Pr(PD)=0,\ Pr(SD)=0,\ Pr(SU)=1\}$
Active Detected $(Z_{11}^i(t))$ On-going Detected $\xrightarrow{\lambda_{S/D}}$ Success Detected $\quad S_t\leftarrow1$	** The detection has occured at a different leaf* *** Despite D and SD having null durations, these lines are necessary to specify the transfer function, the transfer being potentially triggered by the leaf itself.*

13

Table 6: The triggered Markov processes of the Instantaneous Security Event (ISE) leaves
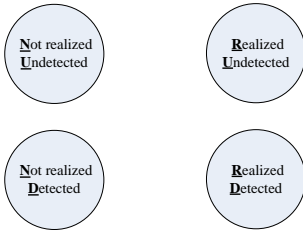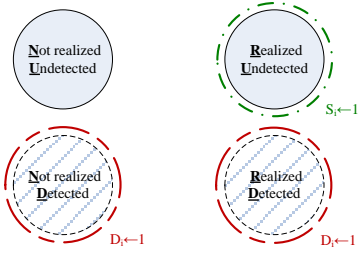
Instantaneous Security Event (ISE)	
Markov processes	**Probability transfer functions**

Idle $(Z_0^i(t))$
 	
 Not realized Undetected   Realized Undetected
 	
 Not realized Detected   Realized Detected
	$f_{0\to10}^i (NU)=\{Pr(NU)=(1-\gamma_{S/ND})(1-\gamma_{D/NR}), Pr(RU)=\gamma_{S/ND}(1-\gamma_{D/R}),$
	         $P(ND)=(1-\gamma_{S/ND})\gamma_{D/NR}, P(RD)=\gamma_{S/ND}\gamma_{D/R}\}$
	    $(RU)=\{Pr(NU)=0, Pr(RU)=1, Pr(ND)=0, Pr(RD)=0\}$***
	    $(ND)=\{Pr(NU)=0, Pr(RU)=0, Pr(ND)=1-\gamma_{S/D}, Pr(RD)=\gamma_{S/D}\}$
	    $(RD)=\{Pr(NU)=0, Pr(RU)=0, Pr(ND)=0, Pr(RD)=1\}$
	
	 $f_{0\to11}^i (NU)=\{Pr(ND)=(1-\gamma_{S/ND}), Pr(RD)=\gamma_{S/ND}\}$*
	    $(RU)=\{Pr(ND)=0, Pr(RD)=1\}$
	    $(ND)=\{Pr(ND)=(1-\gamma_{S/D}), Pr(RD)=\gamma_{S/D}\}$*
	    $(RD)=\{Pr(ND)=0, Pr(RD)=1\}$

Active Undetected $(Z_{10}^i(t))$
 	
 Not realized Undetected   Realized Undetected $S_t\leftarrow1$
 	
 Not realized Detected $D_t\leftarrow1$ Realized Detected $D_t\leftarrow1$
	$f_{10\to11}^i(NU)=\{Pr(ND)=1, Pr(RD)=0\}$*
	    $(RU)=\{Pr(ND)=0, Pr(RD)=1\}$*
	    $(ND)=\{Pr(ND)=1, Pr(RD)=0\}$**
	    $(RD)=\{Pr(ND)=0, Pr(RD)=1\}$**
	
	 $f_{11\to0}^i\ (ND)=\{Pr(NU)=0, Pr(RU)=0, Pr(ND)=1, Pr(RD)=0\}$
	    $(RD)=\{Pr(NU)=0, Pr(RU)=0, Pr(ND)=0, Pr(RD)=1\}$
	
	 $f_{10\to0}^i\ (NU)=\{Pr(NU)=1, Pr(RU)=0, Pr(ND)=0, Pr(RD)=0\}$
	    $(RU)=\{Pr(NU)=0, Pr(RU)=1, Pr(ND)=0, Pr(RD)=0\}$

Active Detected $(Z_{11}^i(t))$ Not realized Detected Realized Detected $S_t\leftarrow1$	*The detection has occured at a different leaf* ** *Despite D and SD having null durations, these lines are necessary to specify the transfer function, the transfer being potentially triggered by the leaf itself.* *** *We assumte that once the leaf is realized, the potential reactivations cannot trigger detection anymore (NB: this differs from the MMM-ACNS paper version).*

Table 7: The triggered Markov process of the Timed Security Event (TSE) leaf

Timed Security Event (TSE)	
Markov processes	**Probability transfer functions**

Idle $(Z_0^i(t))$

$f_{0 \to 10}^i(PU)=\{Pr(NU)= 1 - \gamma_{D(I)}, Pr(ND)= \gamma_{D(I)}, Pr(RD)= 0, Pr(RU)= 0\}$
$(PD)=\{Pr(NU)= 0, Pr(ND)= 1, Pr(RD)= 0, Pr(RU)= 0\}$
$(NU)=\{Pr(NU)= 1, Pr(ND)= 0, Pr(RD)= 0, Pr(RU)= 0\}$
$(RU)=\{Pr(NU)= 0, Pr(ND)= 0, Pr(RD)= 0, Pr(RU)= 1\}$
$(ND)=\{Pr(NU)= 0, Pr(ND)= 1, Pr(RD)= 0, Pr(RU)= 0\}$
$(RD)=\{Pr(NU)= 0, Pr(ND)= 0, Pr(RD)= 1, Pr(RU)= 0\}$

$f_{0 \to 11}^i(PU)=\{Pr(ND)= 1, Pr(RD)= 0\}*$
$(PD)=\{Pr(ND)= 1, Pr(RD)= 0\}$
$(NU)= \{Pr(ND)= 1, Pr(RD)= 0\}*$
$(ND)= \{Pr(ND)= 1, Pr(RD)= 0\}$
$(RD) = \{Pr(ND)= 0, Pr(RD)= 1\}$
$(RU) = \{Pr(ND)= 0, Pr(RD)= 1\}*$

Active Undetected $(Z_{10}^i(t))$

$f_{10 \to 11}^i(NU)= \{Pr(ND)= 1, Pr(RD)= 0\}*$
$(ND)= \{Pr(ND)= 1, Pr(RD)= 0\}**$
$(RD) = \{Pr(ND)= 0, Pr(RD)= 1\}**$
$(RU) = \{Pr(ND)= 0, Pr(RD)= 1\} *$

$f_{11 \to 0}^i(ND)=\{Pr(PU)=0, Pr(PD)=0, Pr(NU)=0, Pr(ND)=1, Pr(RD)=0, Pr(RU)=0\}$
$(RD)=\{Pr(PU)=0, Pr(PD)=0, Pr(NU)=0, Pr(ND)=0, Pr(RD)=1, Pr(RU)=0\}$

$f_{10 \to 0}^i(NU)=\{Pr(PU)=0, Pr(PD)=0, Pr(NU)=1, Pr(ND)=0, Pr(RD)=0, Pr(RU)=0\}$
$(ND)=\{Pr(PU)=0, Pr(PD)=0, Pr(NU)=0, Pr(ND)=1, Pr(RD)=0, Pr(RU)=0\}$
$(RD)=\{Pr(PU)=0, Pr(PD)=0, Pr(NU)=0, Pr(ND)=0, Pr(RD)=1, Pr(RU)=0\}$
$(RU)=\{Pr(PU)=0, Pr(PD)=0, Pr(NU)=0, Pr(ND)=0, Pr(RD)=0, Pr(RU)=1\}$

Active Detected $(Z_{11}^i(t))$

The detection has occured at a different leaf

**Despite D and SD having null durations, these lines are necessary to specify the transfer function, the transfer being potentially triggered by the leaf itself.*

15

of success, just by the off-line bruteforce attack. In terms of sequences analysis, the number of possible sequences is much higher (4231 vs. 656 in Section 3.4). Tab. 9 gives a selection of sequences with the conventions of Tab. 3; in addition, detections that occurred are indicated in brackets for the relevant leaves. Here again, the top 2 sequences are direct successes of social engineering techniques, followed by the success of a direct bruteforce attack. In the present case, they are followed by several bruteforce terminated non-minimal sequences, before the first sequences based on the trapped email with malicious payload approach appear (seq. 14 and 17). This differs from Tab. 3 in which the sequences based on physical approaches appear first, whereas they are relegated to seq. 20 and further in the present case. This is related to the detection and reaction possibilities associated here to such sequences. In seq. 20, the attacker has failed in his social engineering attempt to manipulate the user by a forged email and has been detected; the parameters of the subsequent leaves are those corresponding to a detected status. Here again, a complete analysis is not provided, but would benefit from success sub-sequences consolidation views.

Table 8: Parameters used for detection and reaction modeling

Leaf name	Type	Undetected modes parameter	Detection parameters	Detected modes parameter
User trapped	ISE	$\gamma_{S/ND} = 0.33$	$\gamma_{D/R} = 0$, $\gamma_{D/NR} = 0.5$	Not used
Appropriate payload	ISE	$\gamma_{S/ND} = 0.1$	$\gamma_{D/R} = 0.1$, $\gamma_{D/NR} = 0.33$	$\gamma_{S/D} = 0.1$ (unchanged)
Crafted attachement opened	TSE	$\lambda_{R/ND} = 1.157 \times 10^{-5}\ s^{-1}$ (MTTS $\approx$ 1 day)	No detection possible	$\lambda_{R/D} = 5.787 \times 10^{-6}\ s^{-1}$ (MTTS $\times 2 \approx$ 2 days)
Password intercepted	TSE	$\lambda_{R/ND} = 1.157 \times 10^{-5}\ s^{-1}$ (MTTS $\approx$ 1 day)	No detection possible	$\lambda_{R/D} = 5.787 \times 10^{-6}\ s^{-1}$ (MTTS $\times 2 \approx$ 2 days)
Physical reconnaissance	AA	$\lambda_{S/ND} = 5.787 \times 10^{-6}\ s^{-1}$ (MTTS $\approx$ 2 days)	$\lambda_{D(O)} = 3.858 \times 10^{-6}\ s^{-1}$ (MTTS $\approx$ 3 days), $\gamma_{D(I)}, \gamma_{D(F)}, \lambda_{D(A)} = 0$	$\lambda_{S/D} = 2.893 \times 10^{-6}\ s^{-1}$ (MTTS $\times 2 \approx$ 4 days)
Keylogger local installation	AA	$\lambda_{S/ND} = 1.157 \times 10^{-5}\ s^{-1}$ (MTTS $\approx$ 1 day)	$\lambda_{D(O)} = 3.472 \times 10^{-5}\ s^{-1}$ (MTTS $\approx$ 8 hours), $\gamma_{D(I)}, \lambda_{D(A)} = 0$ $\gamma_{D(F)} = 0.1$	$\lambda_{S/D} = 5.787 \times 10^{-6}\ s^{-1}$ (MTTS $\times 2 \approx$ 2 days)

Table 9: Selection of sequences with quantifications

		Sequences	Probability in a week	Average duration (s)	Contrib.
1		<Social Eng>Generic reconn., Email trap exec., User trapped	1.091×10^{-1}	9.889×10^4	30.0%
2		<Social Eng>Generic reconn., Phone trap exec., User trapped	5.456×10^{-2}	9.889×10^4	15.0%
3		Bruteforce	2.144×10^{-2}	5.638×10^4	5.9%
4		<Social Eng>*Generic reconnaissance*, Bruteforce	1.055×10^{-2}	9.889×10^4	2.9%
...		([...], Bruteforce) $\times$ 9			
14		<Social Eng><Social Eng><Keylogger><Remote>Generic reconnaissance, Payload crafting(no detection), Appropriate payload(no detection), Password intercepted	2.250×10^{-3}	2.761×10^5	0.6%
...		([...], Bruteforce) $\times$ 2			
17		<Social Eng>Generic reconnaissance <Social Eng><Keylogger> <Remote>Payload crafting(no detection), Appropriate payload(no detection), Password intercepted	1.923×10^{-3}	2.688×10^5	0.5%
...		([...], Bruteforce) $\times$ 2			
20		<Social Eng>*Generic reconnaissance, Email trap exec., User trapped(failure and detection)* <Social Eng><Keylogger><Remote><Remote> <Physical>Physical reconn., Keylogger local installation, Password intercepted	1.549×10^{-3}	5.991×10^5	0.4%

5 On-going and Future Work

5.1 Finer and Easier Analyses to Support Security Decision

The new modes related to detection enable new quantifications which may be of interest for the analyst. This includes the mean time to detection (MTTD) or attack sequences classification ordered by their probability of detection. Besides, if the list of sequences provides insightful qualitative and quantitative information, finer-grain analysis, for instance regarding success sub-sequences, are needed to take complete advantage of the model results. Moreover, individual leaf importance factors, adapted to dynamic models as discussed in [20], could be defined for our framework to complete the analyst tool-box. We intend to develop complete and automated tools implementing all these aspects in order to provide a finer and easier support to security decision.

5.2 Non-Markovian Framework

The BDMP theoretical framework has been built on Markovian assumptions and exponential distributions, commonly accepted in reliability engineering [21]. Although such a framework has also been used in security (see [2] for a short review), there is much debate on the appropriate way to model stochastically the behavior of an intelligent attacker, if any. In this perspective, it may be of interest to enable the use of other distributions. This is possible without changing the graphical formalism, but the quantifications could not fully benefit from the methods described in Section 3.4 and would rely on Monte-Carlo simulation.

5.3 BDMP Security Patterns Library

The construction of diverse models during this research has led to the identification of recurrent patterns in attack scenarios. A rigorous inventory and categorization of such patterns could lead to a library of small BDMP, modeling classical attack steps ready to assemble when building a complete model.

6 Conclusion

The adaptation and extension of the BDMP formalism offers a new security modeling technique which combines readability, scalability and quantification capability. This paper has presented a complete view of its mathematical framework and has illustrated its use through different use-cases. Sequences, but also concurrent actions or exclusive choices can be easily taken into account. On the defensive side, detection aspects have been integrated while several alternatives are possible for reaction modeling. This extended formalism inherits from the hierarchical and scalable structure of attack trees, allowing different depths of analysis and ease of appropriation, but goes far beyond by taking into account the dynamics of security. It enables diverse and efficient time-domain quantifications, taking advantage of the BDMP trimming mechanism and their associated sequence exploration approach, which have been used extensively in the reliability engineering area. If there is still room for further developments as seen in Section 5, the framework presented here can be already considered as ready to use, bringing an original approach in the security modeling area.

References

[1] M. Bouissou and J. Bon, "A new formalism that combines advantages of fault-trees and Markov models: Boolean logic driven Markov processes," *Reliability Engineering & System Safety*, vol. 82, pp. 149–163, Nov. 2003.

[2] L. Piètre-Cambacédès and M. Bouissou, "Beyond attack trees: dynamic security modeling with Boolean logic Driven Markov Processes (BDMP)," in *Proceedings of the 8th European Dependable Computing Conference (EDCC)*, (Valencia, Spain), pp. 199–208, Apr. 2010.

[3] M. Bouissou, "Automated dependability analysis of complex systems with the KB3 workbench: the experience of EDF R&D," in *Proceedings of the International Conference on Energy and Environment (CIEM'05)*, (Bucharest, Romania), Oct. 2005.

[4] B. Schneier, "Attack trees: Modeling security threats," *Dr. Dobb's*, vol. 12, no. 24, pp. 21–29, 1999.

[5] S. Mauw and M. Oostdijk, "Foundations of attack trees," in *Proceedings of the 8th Annual Int. Conf. on Information Security and Cryptology (ICISC'05), LNCS 3935*, (Seoul, Korea), pp. 186–198, Dec. 2005.

[6] E. G. Amoroso, *Fundamentals of computer security technology*, ch. 2: Threat Trees, pp. 15–29. Prentice-Hall Inc., USA, 1994.

[7] S. C. Patel, J. H. Graham, and P. A. Ralston, "Quantitatively assessing the vulnerability of critical information systems: A new method for evaluating security enhancements," *International Journal of Information Management*, vol. 28, pp. 483–491, Dec. 2008.

[8] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. Wing, "Automated generation and analysis of attack graphs," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P'02)*, (Oakland, USA), pp. 273–284, May 2002.

[9] R. Lippmann and K. Ingols, "An annotated review of past papers on attack graphs," Project Report ESC-TR-2005-054, Massachusetts Institute of Technology (MIT), Lincoln Laboratory, Mar. 2005.

[10] I. Kotenko and M. Stepashkin, "Analyzing network security using malefactor action graphs," *International Journal of Computer Science and Network Security*, vol. 6, no. 6, pp. 226–236, 2006.

[11] J. P. McDermott, "Attack net penetration testing," in *Proceedings of the 2000 Workshop on New Security Paradigms (NSPW'00)*, (Cork, Ireland), pp. 15–21, Sept. 2000.

[12] S. Pudar, G. Manimaran, and C. Liu, "PENET: a practical method and tool for integrated modeling of security attacks and countermeasures," *Computers & Security*, vol. 28, pp. 754–771, May 2010.

[13] D. M. Nicol, W. H. Sanders, and K. S. Trivedi, "Model-based evaluation: From dependability to security," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 48–65, 2004.

[14] L. Piètre-Cambacédès and C. Chaudet, "Disentangling the relations between safety and security," in *Proceedings of the 9th WSEAS International Conference on Applied Informatics and Communications (AIC'09)*, (Moscow, Russia), pp. 156–161, Aug. 2009.

[15] M. Bouissou and Y. Lefebvre, "A path-based algorithm to evaluate asymptotic unavailability for large Markov models," in *Proceedings of the 48th Reliability and Maintainability Annual Symposium (RAMS'02)*, (Seattle, USA), pp. 32–39, 2002.

[16] P. Harrison, "Laplace transform inversion and passage time distributions in Markov processes," *Journal of Applied Probability*, vol. 27, no. 1, pp. 74–87, 1990.

[17] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid, and D. Gollmann, "Towards operational measures of computer security," *Journal of Computer Security*, vol. 2, pp. 211–229, 1993.

[18] E. Jonsson and T. Olovsson, "A quantitative model of the security intrusion process based on attacker behavior," *IEEE Transactions on Software Engineering*, vol. 23, no. 4, pp. 235–245, 1997.

[19] K. Sallhammar, *Stochastic models for combined security and dependability evaluation*. PhD thesis, Norwegian University of Science and Technology NTNU, 2007.

[20] Y. Ou and J. B. Dugan, "Approximate sensitivity analysis for acyclic Markov reliability models," *IEEE Transactions on Reliability*, vol. 52, pp. 220–230, June 2003.

[21] M. Rausand and A. Høyland, *System Reliability Theory*. Wiley, 2nd ed., 2004.