évaluation quantitative de la sécurité. Approche basée sur les vulnérabilités

Géraldine Vache Marconato, Vincent Nicomette, Mohamed Kaâniche

Evaluation quantitative de la sécurité : approche basée sur les vulnérabilités

Géraldine Vache Marconato[1,2] — Vincent Nicomette[1,3] — Mohamed Kaâniche[1,2]

[1] *CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France*
[2] *Univ. Toulouse, LAAS, F-31400 Toulouse, France*
[3] *Univ. Toulouse, INSA, LAAS, F-31400 Toulouse, France*

{gvache,nicomett,kaaniche}@laas.fr

RÉSUMÉ. *Les travaux présentés dans cet article sont consacrés à la sécurisation des systèmes d'information en se basant sur des mesures quantitatives. Les mesures ont pour but d'aider à la prévision des risques et de fournir les informations nécessaires permettant d'assurer le niveau de sécurité du système en opération. Dans cette approche, nous prenons en considération des facteurs externes ayant un impact significatif sur la sécurité du système. Nous avons identifié trois facteurs, liés au processus d'exploitation d'une vulnérabilité : le cycle de vie de la vulnérabilité, le comportement des attaquants et le comportement de l'administrateur du système. Nous avons étudié les dépendances entre ces différents facteurs et comment leur évolution pouvait impacter la sécurité du système. A partir de cette étude, nous avons défini des mesures quantitatives prenant en compte ces facteurs externes et avons modélisé, en utilisant le formalisme des SAN (Stochastic Activity Networks), le processus d'exploitation de la vulnérabilité qui conduit à la compromission du système. Nous avons distingué deux scénarios selon que la vulnérabilité est découverte par une personne malveillante ou non malveillante. En analysant une base de données de vulnérabilités, nous avons caractérisé la probabilité d'occurrence de plusieurs événements du cycle de vie de la vulnérabilité, que nous utilisons pour l'obtention de nos mesures.*

ABSTRACT. *The objective of this work is the evaluation of information systems security using quantitative measures. These measures aim at forecasting risks and providing information to monitor the security level of the system in operation. In our approach, we take into account some environmental factors that have a significant impact on the security of the system. We have identified three such factors that are related to the vulnerability exploitation process: the vulnerability life cycle, the behavior of the attackers and the behavior of the system administrator. We have studied the dependencies between these factors and how the evolution of these factors could impact the system security. From this study, we have defined quantitative security measures taking into account these environmental factors and we have developed a model*

1re soumission à *Technique et science informatiques*, le 22/03/2011

based on Stochastic Activity Networks (SANs), describing how the vulnerability exploitation process could lead to system compromission. We have distinguished two scenarios according to whether the vulnerability is discovered by a malicious user or not. By analysing a vulnerability database, we have characterised the probability of occurrence of several events of the vulnerability life cycle. This characterization helped us to quantify the measures by processing the SAN model.

MOTS-CLÉS : *sécurité informatique, évaluation, vulnérabilité, mesures, modèles stochastiques*

KEYWORDS: *security, evaluation, vulnerability, metrics, stochastic models*

1. Introduction

Assurer la sécurité d'un système d'information est un problème aussi crucial que complexe : depuis 2006, plus de 7000 vulnérabilités ont été publiées chaque année. Dans ce contexte, évaluer la sécurité d'un système d'information apparaît comme indispensable pour anticiper et analyser les risques.

Les premières approches pour l'évaluation de la sécurité sont apparues dans les années 80 avec les critères d'évaluation tels que les TCSEC (U.S. Department Of Defense, 1985), les ITSEC (European Communities, 1991) et plus récemment les Critères Communs (ISO/CEI 15408, 1996). Ils définissent des niveaux de sécurité, les moyens et processus qui doivent être mis en oeuvre dès la conception pour satisfaire ces niveaux de protection. Les mesures définies dans ces critères et standards sont considérées comme étant des mesures qualitatives, même si la frontière entre évaluation qualitative et quantitative reste floue dans le domaine de la sécurité. Les standards de la famille ISO 27000, tels que (ISO/IEC 27001, 2005; ISO/IEC 27002, 2005), définissent des niveaux de sécurité en tenant compte des fonctionnalités implémentées et du degré de rigueur et de formalisation du processus de développement, considérés également comme des mesures qualitatives. Malheureusement, ces critères d'évaluation n'ont pas été conçus pour évaluer les risques pour la sécurité du système en considérant un environnement dynamique : les processus d'évaluation prennent trop de temps pour être exécutés régulièrement durant la vie opérationnelle du système.

Notre approche se concentre sur ce problème et a pour objectif de définir des mesures quantitatives pour évaluer le niveau de risques encouru par un système en opération en considérant les changements de l'environnement. Ces mesures ont pour objectif de permettre d'évaluer et de quantifier l'impact de différents facteurs externes sur la sécurité du système. Dans ce but, nous identifions les facteurs externes qui ont un impact sur le processus d'exploitation d'une vulnérabilité : le cycle de vie de la vulnérabilité et deux facteurs environnementaux qui sont 1) le comportement de la population des attaquants et 2) le comportement de l'administrateur. Afin de fournir des mesures et de quantifier plusieurs aspects de la sécurité, nous étudions l'évolution de ces trois facteurs, définissons des mesures considérant leur impact et modélisons ces facteurs et leurs interactions avec le système. Puis nous quantifions les probabilités d'occurrence de certains événements du cycle de vie de la vulnérabilité et exécutons le modèle que nous avons développé pour évaluer les conséquences de ces facteurs sur la sécurité du système. Nous évaluons en particulier la probabilité que le système soit compromis ou se trouve dans des états qui présentent un danger vis-à-vis de la sécurité.

Cet article est structuré comme suit : la section 2 décrit les travaux existants sur l'évaluation quantitative de la sécurité. La section 3 présente en détail les trois facteurs que nous considérons et étudie leur impact sur le système. Les sections 4 et 5 sont dédiées, respectivement, à la définition des mesures et à la description du modèle permettant leur évaluation. La section 6 est consacrée à l'estimation de la probabilité d'occurrence d'événements du cycle de vie de la vulnérabilité basés sur les données

fournies par une base de données de vulnérabilités. La section 7 détaille les mesures quantitatives et les résultats obtenus à partir des modèles. Enfin, la section 8 conclut ce papier et présente nos perspectives.

2. Contexte et problématique

Dans cette section, nous présentons différents travaux existants sur l'évaluation quantitative de le sécurité. Nous discutons d'abord de la mesure même de la sécurité et des différents types d'évaluation qui ont été développés dans le domaine de la sécurité. Nous présentons ensuite un bref aperçu de l'état de l'art dans ce domaine.

2.1. *Evaluation et mesure de la sécurité*

L'évaluation de la sécurité a pour objectif, d'une part, d'identifier les vulnérabilités et les menaces susceptibles d'affecter le système et leurs caractéristiques, et d'autre part, d'analyser voire quantifier leur impact vis-à-vis des propriétés attendues (confidentialité, intégrité, disponibilité, etc.).

L'évaluation nécessite généralement la définition de mesures, ou métriques, qui doivent être rattachées à une échelle de mesure pour avoir une signification exploitable (Zuse, 1991). Il existe plusieurs types d'échelle : les échelles nominales, les échelles ordinales, les échelles de rapport et les échelles absolues. Par exemple, l'échelle nominale est une énumération non ordonnée qui permet de classer sans autoriser aucune opération entre les classes ainsi constituées. Dans le cas d'une échelle ordinale, en revanche, les catégories qui la composent sont munies d'une structure d'ordre, permettant d'établir des comparaisons entre les différentes catégories.

Une mesure peut possèder, ou non, une unité. Il est important avant tout de définir l'objectif de la mesure et la réponse que doit fournir l'évaluation ? Une réponse de type OUI/NON peut suffire à répondre à une question telle que "mon système est-il suffisamment protégé pour garantir ma politique de sécurité ?". Mais une réponse plus poussée peut être souhaitée, en considérant par exemple, des questions du type : "Je veux garantir ou atteindre un niveau de sécurité donné pour mon système. Quels sont les efforts et les moyens qui devront être déployés ? ", ou bien : "Mon système possède tels moyens de protection, quels risques encourt-il vis-à-vis de la sécurité ?"

De nombreux travaux ont cherché à répondre à ce type de questions. On peut distinguer trois types d'approches d'évaluation qui sont complémentaires : 1) ordinales basées sur des critères qualitatifs tels que les critères communs ou les ITSEC qui sont généralement focalisés sur les processus et les fonctions à mettre en oeuvre en fonction du niveau d'assurance recherché , 2) quantitatives ou probabilistes à base de modèles permettant de prendre en compte différent types d'incertitudes concernant les vulnérabilités et attaques potentielles ainsi que l'efficacité des moyens de protection mis en oeuvre, ou bien 3) expérimentales basées sur l'observation de systèmes en conditions

opérationnelles ou dans le cadre d'expériences contrôlées, ainsi que sur la collecte d'informations sur les vulnérabilités et attaques connues.

Les travaux présentés dans ce papier focalisent sur les approches probabilistes à base de modèles. Nous présentons dans la section 2 une discussion de l'état de l'art sur l'évaluation de la sécurité afin de mieux situer nos contributions. Notre approche d'évaluation et les mesures considérées sont discutées dans les sections 4 et 5.

2.2. *Etat de l'art sur l'évaluation de la sécurité*

Plusieurs travaux ont été proposés pour permettre une évaluation quantitative de la sécurité durant la vie opérationnelle du système. Dans (Nicol *et al.*, 2004), les auteurs mettent en évidence les besoins de techniques d'évaluation pour la sécurité et comparent les travaux et méthodologies déjà existants. En 1993, dans (Jonsson *et al.*, 1997), l'auteur affirme que la sécurité peut être évaluée en terme d'effort, mais sans proposer de mesure ou de méthodologie pour ce type d'évaluation. Durant la même année, dans (Dacier, 1994; Dacier *et al.*, 1996), les auteurs proposent le formalisme du graphe des privilèges, qui met en évidence les différents chemins d'exploitation de vulnérabilités qu'un attaquant peut suivre pour acquérir de facon malveillante certains privilèges. Un privilège se définit comme un ensemble de droits qu'un utilisateur, ou un groupe d'utilisateurs, peut posséder sur un objet. Chaque noeud du graphe représente un ensemble de privilèges. Le graphe des privilèges est utilisé pour obtenir la mesure METF (*Mean Effort To security Failure*) qui a pour objectif de caractériser la capacité du système à résister à des attaques (Ortalo *et al.*, 1999).

Le formalisme des graphes d'attaque décrit dans (Sheyner, 2004) est basé sur des concepts similaires : chaque état du graphe représente les privilèges possédés par l'attaquant ainsi que la connaissance de l'attaquant et l'état de l'environnement du système. Plusieurs études ont été menées sur la génération et l'optimisation de ces graphes, par exemple (Sheyner, 2004; Jha *et al.*, 2002; Swiler *et al.*, 2001). L'arbre d'attaque est un autre formalisme qui peut être utilisé pour décrire des scénarios d'attaque et également pour obtenir des mesures quantitatives de la sécurité, comme par exemple, la mesure d'exploitabilité définie dans (Balzarotti *et al.*, 2006) : une mesure d'exploitabilité pondère chaque arc de l'arbre.

Une autre mesure appelée "Time To Compromise" est présentée dans (McQueen *et al.*, 2006). Elle est basée sur trois différents processus correspondant à trois situations pour l'attaquant : 1) l'attaquant connait au moins une vulnérabilité qui lui donne les privilèges voulus et il existe au moins un *exploit* connu ; 2) il existe au moins une vulnérabilité connue dont l'exploitation permet d'obtenir les privilèges voulus et l'attaquant ne connaît pas d'attaque disponible et recherche donc une attaque qui exploiterait avec succès cette vulnérabilité ; 3) l'attaquant cherche à identifier de facon permanente de nouvelles vulnérabilités puis des attaques exploitant celles-ci. Les processus 1 et 2 sont exclusifs et concernent l'exploitation des vulnérabilités déjà connues. Le processus 2 est exécuté seulement si le processus 1 échoue et si les conditions de dé-

part de ce processus ne sont plus valides. Le processus 3 est en exécution permanente et parallèle aux processus 1 et 2. La mesure "Time To Compromise" résultante dépend donc de la probabilité d'occurrence de ces processus et du temps nécessaire à la réussite d'une l'attaque. Cette mesure nécessite de connaître le nombre de vulnérabilités présentes dans le système étudié et donne des mesures dépendant d'un seul type d'attaquant.

L'attaquant n'est pas le seul facteur externe à pouvoir avoir un impact sur la sécurité du système. Dans (Mell *et al.*, 2007), trois mesures, considérant plusieurs facteurs externes, sont présentées : 1) la mesure de "base", tenant compte des droits d'accès nécessaires pour exploiter la vulnérabilité ainsi que de l'impact sur les attributs de la sécurité ; 2) une mesure temporelle tenant compte de l'existence d'un *exploit* et d'un correctif ; 3) une mesure d'environnement tenant compte de l'impact d'une attaque sur l'environnement physique du système. Des équations sont proposées pour obtenir les valeurs quantitatives associées à ces mesures mais les valeurs des coefficients ne sont pas explicitement justifiées.

Ces mesures quantitatives ont pour objectif d'évaluer la sécurité du système en opération et de prendre en considération son environnement sans toutefois tenir compte en même temps du cycle de vie de la vulnérabilité. D'autres études se sont intéressés à l'utilisation de données réelles afin de caractériser des comportements d'attaquant. Par exemple, dans (Arora *et al.*, 2004), les auteurs étudient l'impact de la publication de la vulnérabilité et de la publication du correctif sur le processus d'attaque. En utilisant un ensemble de 308 vulnérabilités, ils ont proposé un modèle permettant de prédire l'évolution du nombre d'attaques par hôte et par jour. Dans (Frei *et al.*, 2006), les analyses se sont concentrées sur la date de publication du correctif et la disponibilité d'un *exploit* en prenant la date de publication de la vulnérabilité comme origine. La date à partir de laquelle la vulnérabilité a été découverte n'est pas prise en compte. Cette étude inclut 14 326 vulnérabilités collectées dans différentes bases de données de vulnérabilités.

Dans (Ozment *et al.*, 2006), les auteurs analysent 140 vulnérabilités issues du système d'exploitation OpenBSD. Il s'agit d'une extension des travaux menés dans (Rescorla, 2005), qui étudie le cycle de vie de la vulnérabilité et conclut que le taux de découverte de vulnérabilités peut être considéré comme constant pour un système d'exploitation donné.

L'approche par modélisation et les résultats que nous présentons dans ce papier abordent à la fois l'étude du processus d'attaque et l'utilisation de données pour permettre de prendre en considération à la fois l'environnement du système, le cycle de vie de la vulnérabilité en fournissant des mesures basées sur des données réelles. Nous considérons que la probabilité d'occurrence d'une attaque exploitant une vulnérabilité n'est pas constante dans le temps : la probabilité qu'un attaquant exploite une vulnérabilité pour laquelle il n'existe pas encore de correctif peut être supérieure à la probabilité qu'il exploite une ancienne vulnérabilité, dont le correctif est disponible depuis longtemps, en supposant que l'attaquant possède une connaissance suffisante ou un *exploit* pour arriver à ses fins. L'impact de l'exploitation de la vulnérabilité sur

le système dépend donc de l'évolution de l'environnement, comme cela est présenté dans la section suivante.

3. Les facteurs externes

Notre objectif est double : 1) produire des mesures quantitatives tenant compte des trois facteurs externes ayant un impact sur le sécurité du système (le cycle de vie de la vulnérabilité, le comportement des attaquants et le comportement de l'administrateur) ; 2) étudier comment un changement de l'environnement peut avoir un impact sur l'évolution de la probabilité pour le système d'être *sûr* ou *compromis*. Bien sûr, l'environnement du système peut être varié : il est évident qu'un système d'information militaire n'aura pas le même environnement qu'un système d'information bancaire. Dans notre approche, nous nous intéressons aux systèmes d'information grand public. Dans cette section, nous identifions les facteurs externes importants et étudions comment ces facteurs interagissent entre eux et sur le système.

3.1. *Le cycle de vie de la vulnérabilité*

Nous définissons le cycle de vie de la vulnérabilité comme l'ensemble d'événements pouvant avoir lieu durant la vie de la vulnérabilité. Dans (Arbaugh *et al.*, 2000), les auteurs tiennent compte des événements de découverte de la vulnérabilité, de la publication de la vulnérabilité, de la publication du correctif, de la disponibilité de l'*exploit*, mais aussi de la naissance et de la mort de la vulnérablité. Cette approche est également suivie par (Rescorla, 2005), en considérant que la disponibilité de l'*exploit* et les attaques résultant de l'utilisation de cet *exploit* ont toujours lieu après la publication de la vulnérabilité. Dans (Frei, 2009), l'auteur ne prend pas en compte la naissance et la mort de la vulnérabilité mais ajoute l'application du correctif comme un événement du cycle de vie de la vulnérabilité. Dans notre approche, nous avons pour objectif de caractériser de manière quantitative les événements du cycle de vie de la vulnérabilité. L'application du correctif n'est pas incluse dans notre définition du cycle de vie mais est prise en considération dans la caractérisation du comportement de l'administrateur (cf. Section 3.3). Nous nous focalisons donc sur les principaux événements du cycle de vie de la vulnérabilité, qui sont considérés dans la majorité des approches et que nous définissons comme suit :

– la découverte de la vulnérabilité : c'est l'instant à partir duquel une personne a connaissance de l'existence de la vulnérabilité ;

– la publication de la vulnérabilité : c'est l'instant de la première apparition de la vulnérabilité sur un canal reconnu où l'information sur la vulnérabilité est librement accessible et a fait l'objet d'une analyse par des experts ;

– la publication du correctif de la vulnérabilité : c'est l'instant à partir duquel il existe un correctif disponible librement permettant de masquer ou de corriger la vulnérabilité ;

– l'apparition de l'*exploit* : elle permet à l'ensemble de la population d'attaquants de perpétrer une attaque exploitant la vulnérabilité. Le kit d'exploitation peut être mis au point par la population des attaquants ou résulter du détournement d'une preuve de concept publiée en même temps que la vulnérabilité.

La disponibilité de l'*exploit* a un impact important sur le comportement des attaquants. Nous pouvons distinguer deux scénarios.

Dans le premier scénario, la vulnérabilité est découverte par une personne non malveillante. Cette personne informe le développeur du composant vulnérable de l'existence de la vulnérabilité et c'est cette action qui va provoquer l'événement de publication de la vulnérabilité. La publication de la vulnérabilité permet aux administrateurs d'être vigilants mais informe également la population malveillante de l'existence de la vulnérabilité. A partir de cet instant, un attaquant peut développer un *exploit*, permettant à l'ensemble de la population des attaquants de perpétrer des attaques.

Dans le second scénario, la vulnérabilité est découverte par une personne malveillante. Cette personne peut ainsi informer d'autres personnes malveillantes de l'existence de la vulnérabilité ou mettre au point elle-même un *exploit*. Ce sont par la suite les attaques perpétrées par l'utilisation de cet *exploit* qui amèneront à l'événement de publication de la vulnérabilité, permettant aux administrateurs de se montrer vigilants jusqu'à la publication du correctif.

Dans ces deux scénarios, nous considérons que le correctif de la vulnérabilité peut être publié après la vulnérabilité ou simultanément. A partir de cette section, le scénario de découverte non malveillante (respectivement malvaillante) sera noté par l'abbréviation NM-S (respectivement M-S).

3.2. *Le comportement des attaquants*

La population des attaquants est un facteur environnemental très important. Cependant, son manque d'homogénéité la rend très difficile à caractériser. Dans (CERIAS, 2005; Rogers, 2006), les auteurs proposent une classification à deux dimensions considérant les motivations et les compétences des attaquants. Dans (Alata *et al.*, 2006), les auteurs étudient deux catégories d'attaquants : les *script kiddies* et les *black hats*. Les premiers sont décrits comme des attaquants "amateurs" ayant besoin de kits d'exploitation pour perpétrer les attaques. Les seconds sont des experts, bien souvent à l'origine des outils utilisés par les premiers. De plus, les auteurs indiquent que les *script kiddies* représentent une très grande proportion de la population des attaquants. Dans ce papier, nous nous focalisons principalement sur cette population d'attaquants.

3.3. *Le comportement de l'administrateur*

Le troisième facteur que nous étudions est le comportement de l'administrateur du système, et le niveau de rigueur avec lequel il gère la sécurité du système d'information. En effet, le fait que l'administrateur soit conscient de ce type de risques ou non peut avoir de nombreuses conséquences sur le système : si l'administrateur ne vérifie pas régulièrement s'il est nécessaire d'appliquer des correctifs et ne les installe pas dès leur publication, le système reste vulnérable longtemps, malgré l'existence d'un correctif disponible. Il est à noter que l'impact du comportement de l'administrateur sur la sécurité du système dépend également du cycle de vie de la vulnérabilité. Dans notre approche, nous considérons le cas pessimiste où il est nécessaire pour l'administrateur d'avoir un correctif disponible pour pouvoir corriger et protéger son système.

Dans la suite de cet article, nous considérerons deux comportements d'administrateurs possibles : un laxiste et un rigoureux. Nous analyserons aussi dans la section 7.2.3 le cas d'un administrateur ayant un comportement intérmédiaire entre ces deux cas extrêmes.

4. Etats du système et définitions des mesures

4.1. *Etats du système*

Dans cette section, nous présentons les différents états du système, en considérant une vulnérabilité ainsi que les facteurs externes que nous avons présentés dans la section 3. Les états définis du système, ainsi que l'impact des facteurs externes considérés sont illustrés dans la figure 1.

Une fois que la vulnérabilité est dans le système, celui-ci devient *vulnérable*. Lorsque le système est dans cet état, cela signifie qu'il n'existe pas encore d'*exploit* disponible. Quand un *exploit* devient disponible, le système devient *exposé*. Il n'y a pas de changement du système à proprement parler puisque ce changement d'état résulte d'un changement dans l'environnement du système.

Dès qu'un *exploit* existe et est disponible à l'ensemble de la population des attaquants, le système peut être victime d'une *attaque avec succès* et ainsi être *compromis*. Le système reste dans ce dernier état jusqu'à ce que l'administrateur applique le correctif de la vulnérabilité sur son système. Cependant, cela ne signifie pas qu'il n'y a plus aucun risque. En effet, un attaquant peut avoir obtenu des nouveaux privilèges grâce à l'exploitation de la vulnérabilité lui permettant de garder un accès au système, même après l'application du correctif, et ce jusqu'à ce que le système ait été proprement assaini. Nous considérons que, dans les deux états *compromis* et *corrigé*, le système est *en danger*. Une fois le correctif appliqué et le système assaini en cas de compromission, le système devient *sûr*. Il est évident que si le correctif existe, il est possible que l'administrateur ait pu corriger le système avant l'apparition d'un *exploit* ou avant qu'un attaquant ait pu mener une attaque à bien. Dans ce cas, l'état du système passe directement de *vulnérable* ou *exposé* à l'état *sûr*.

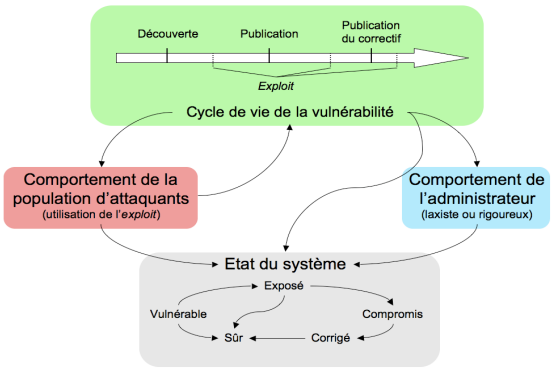

Figure 1. *Résumé des facteurs externes et des états du système*

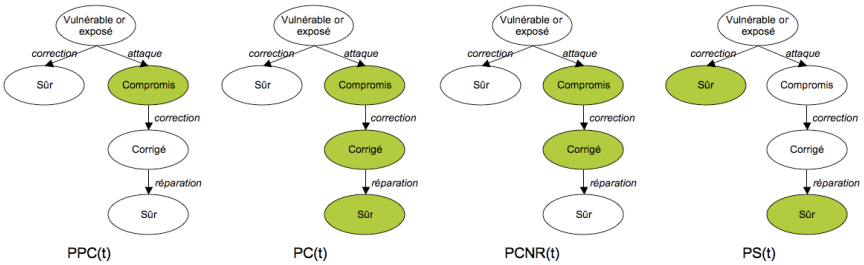

Figure 2. *Mesures*

Différentes mesures peuvent être définies et basées sur les états ou les événements, comme cela est détaillé dans la section suivante. Comme nous venons de le présenter, la différence entre les états *vulnérable* et *exposé* est uniquement causée par un changement dans l'environnement. Pour rendre la définition de nos mesures plus facile à comprendre, nous avons choisi de regrouper ces deux états en un seul nommé *vulnérable ou exposé*. Dans la suite, nous supposons que le système se trouve initialement dans cet état et obtenons différentes mesures caractérisant le passage du système dans les autres états décrits dans la figure 1.

4.2. *Définition des mesures*

Dans notre contexte, nous définissons quatre mesures quantitatives, illustrées par la figure 2. Pour chacune des mesures, cette figure décrit les états du système qui

sont considérés (indiqués par les états colorés) : nous évaluons la probabilité pour le système d'être dans un des états colorés. Les mesures *PPC(t)* et *PCNR(t)* ont pour objectif d'apporter une information sur la probabilité pour le système d'être dans un état où il court un danger immédiat. C'est le cas lorsque le système est dans l'état *compromis*. Cependant, corriger la vulnérabilité en elle-même peut ne pas être suffisant si l'exploitation de celle-ci a permis à l'attaquant d'obtenir des privilèges supplémentaires. C'est pourquoi la mesure *PCNR(t)* prend en considération l'état du système où la vulnérabilité a été corrigée mais le système n'a pas encore été réparé. Les mesures *PC(t)* et *PS(t)* ont un autre objectif qui est de pouvoir évaluer les probabilités pour le système de devenir sûr. Ces deux mesures permettent d'envisager à elles deux tous les scénarios possibles pour atteindre cet état. La mesure *PC(t)* tient compte du scénario le plus critique, à savoir celui dans lequel le système a été compromis. La mesure *PS(t)* adopte une approche plus générale dans laquelle seul l'état sûr est considéré, quelque soit le scénario. Ces mesures sont présentées une par une dans la suite.

– **Probabilité pour le système d'être dans l'état compromis : PPC(t)**

Cette mesure est basée la probabilité que le système soit dans l'état *compromis* à un instant *t*. Cela signifie que le système a été compromis grâce à une attaque perpétrée avec succès. Cette mesure est définie comme suit :

PPC(t) = p(système dans l'état Compromis à l'instant t)

Cette mesure peut également être définie en considérant l'occurrence des événements d'attaque et d'application du correctif :

PPC(t) = p{attaque avec succès et système non corrigé durant [0, t]}

– **Probabilité que le système ait été compromis : PC(t)**

Cette mesure permet de quantifier la probabilié, à un instant *t*, que le système ait été compromis grâce à une attaque perpétrée avec succès. Elle correspond à la probabilité pour le système d'être dans l'état *compromis*, *corrigé* ou *sûr* sachant que le système a été compromis puis réparé. Elle est définie comme suit :

PC(t) = p{système dans l'état (Compromis ou Corrigé ou Sûr/attaque perpétrée avec succès) à l'instant t}

Cette mesure peut être définie également comme suit :

PC(t) = p{attaque perpétrée avec succès durant [0, t]}

Cette mesure prend en considération l'état courant du système mais également les événements antérieurs. Cela permet d'évaluer l'intervalle de temps durant lequel la probabilité que le système soit compromis par l'exploitation d'une vulnérabilité ne dépasse pas un certain seuil.

– **Probabilité de compromission et non réparation : PCNR(t)**

Cette mesure permet de quantifier la probabilité, à l'instant *t*, que le système ait été compromis suite à une attaque perpétrée avec succès et que les dégâts éventuels ne soient pas encore réparés. Elle est définie comme suit :

PCNR(t) = p{système dans l'état Compromis ou Corrigé à l'instant t}

En considérant non plus les états mais les événements, la mesure *PCNR(t)* peut se définir comme suit :

PCNR(t) = p{attaque perpétrée avec succès et non réparation durant [0, t]}

– **Probabilité d'avoir un système sûr : PS(t)**

Cette dernière mesure permet de quantifier, à un instant *t*, la probabilité que le système soit sûr, en tenant compte de l'impact de la vulnérabilité considérée.

PS(t) = p{système dans l'état Sûr à l'instant t}

En considérant les événements plutôt que les états du système, la mesure *PS(t)* peut se définir comme suit :

PS(t) = p{(vulnérabilité corrigée avant attaque) ou (attaque perpétrée avec succès et correction et réparation) durant [0, t]}

Pour obtenir ces mesures, il est nécessaire de modéliser les facteurs que nous avons présentés dans la section 3 et leur impact sur les états du système. La section suivante présente le modèle.

5. Modélisation

Dans cette section, nous présentons notre modéle décrivant l'évolution de l'état du système en tenant compte des facteurs externes. Le modèle peut être utilisé pour obtenir les mesures quantitatives caractérisant les probabilités associées à chacun des états du système, présenté dans la section 4. Une version préliminaire de ce modèle a été présentée dans (Vache, 2009a).

5.1. *Choix du formalisme de modélisation*

Le modèle se base sur le formalisme des SAN (*Stochastic Activity Networks*) (Sanders *et al.*, 2002). Ce formalisme peut être facilement utilisé pour décrire l'évolution de l'état du système et exprimer les conditions d'occurrence des événements caractérisés par différentes distributions de probabilités. Notre choix est aussi justifié par l'existence d'un outil supportant ce formalisme (l'outil Möbius (Daly *et al.*, 2000)) qui est maintenant largement utilisé dans la communauté, y compris dans le domaine de la sécurité (LeMay *et al.*, 2011).

Les SAN se composent de quatre types d'éléments :

– **places** : elles contiennent un ou plusieurs jetons et modélisent les états du système et de l'environnement ;

– **activités** : elles modélisent les événements qui ont un effet sur le système ou son environnement ; l'occurrence de ces événements peut être décrite par des lois probabilistes ou déterministes ;

– **portes d'entrée** : elles contiennent les pré-conditions nécessaires au déclenchement d'une activité en fonction du marquage d'une ou plusieurs autres places ;

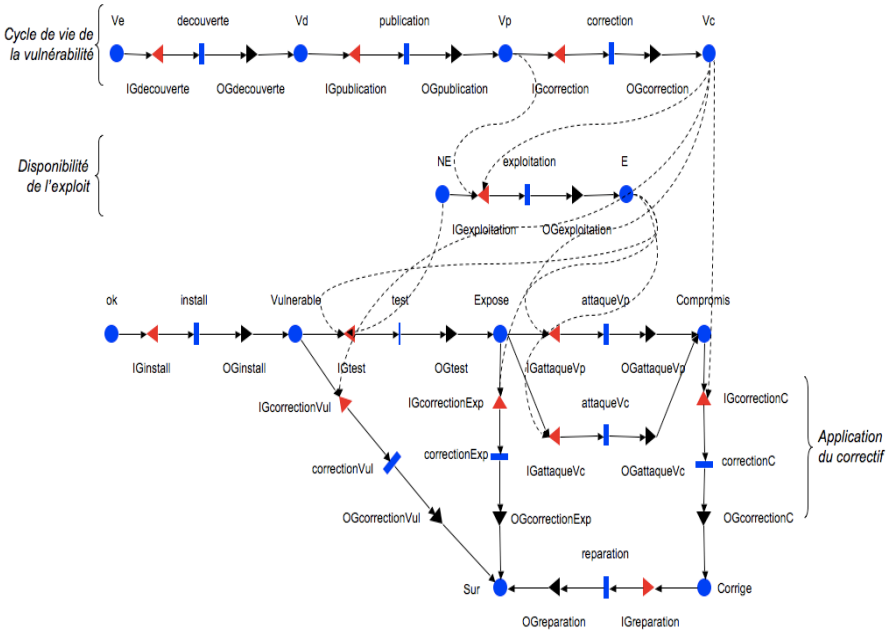

Figure 3. *Modèle du processus d'exploitation de la vulnérabilité : NM-S*

– **portes de sortie** : elles contiennent les conséquences sur le marquage des places du modèle suite au franchissement d'une activité.

Dans la section suivante, nous décrivons les modèles SAN considérant une vulnérabilité.

5.2. *Description des modèles SAN*

Nous avons créé deux modèles, considérant les deux scénarios décrits dans la section 3 et dépendant de l'origine de la découverte de la vulnérabilité, malveillante ou non (cf. figures 3 and 4). Ces modèles sont composés de deux parties : le cycle de vie de la vulnérabilité est modélisé en haut ; le reste du modèle décrit les différents états du système en considérant les comportements des attaquants et de l'administrateur. Les lignes pointillées indiquent les places utilisées dans les prédicats contenus dans les portes d'entrée du modèle. Dans cette section, nous décrivons le modèle plus en détail, en commencant par le cycle de vie de la vulnérabilité.

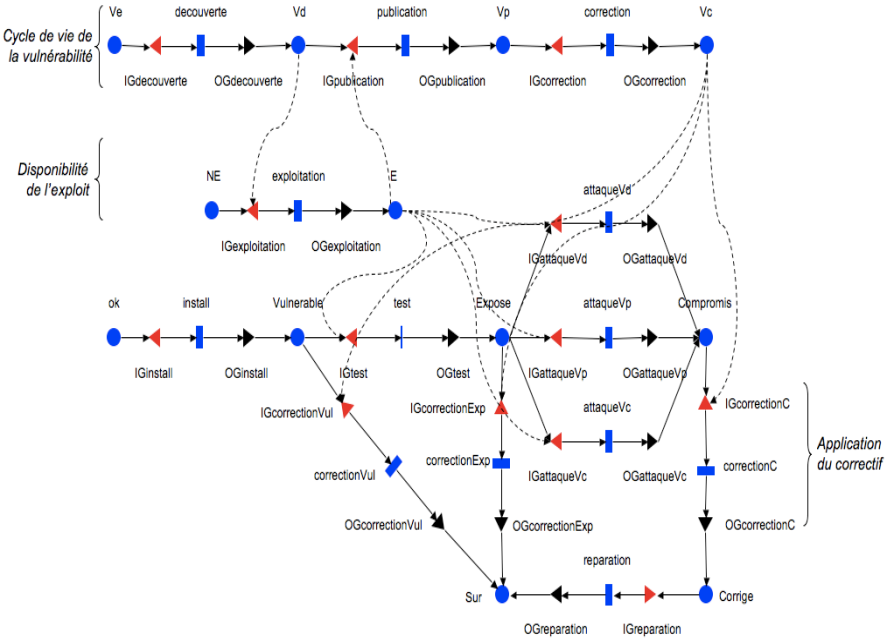

Figure 4. *Modèle du processus d'exploitation de la vulnérabilité : M-S*

5.2.1. *La modélisation du cycle de vie de la vulnérabilité*

Les trois activités (*découverte, publication, correction*) modélisent les trois événements correspondant à la découverte de la vulnérabilité, sa publication et la publication du correctif. Les états entre ces événements sont modélisés par quatre places `Ve`,`Vd`, `Vp`, `Vc` définies comme suit : 1) `Ve` (pour "existence") modélise l'état du cycle de vie tel que la vulnérabilité *existe* mais n'a pas encore été découverte ; 2) `Vd` (pour "découverte") modélise l'état tel que la vulnérabilité a été *découverte* mais n'a pas encore été *publiée* ; 3) `Vp` (pour "publication") modélise l'état dans lequel la vulnérabilité a été *découverte* et *publiée* mais pour laquelle il n'existe pas encore de correctif disponible ; 4) `Vc` (pour "correction") modélise l'état dans lequel la vulnérabilité a été découverte, publiée et pour laquelle il existe un correctif disponible.

La disponibilité d'un *exploit* est modélisée par une activité avec différentes conditions selon le scénario considéré. Ainsi, l'activité nommée `exploit` et décrite dans la figure 3, modélise la disponibilité d'un *exploit* après la publication de la vulnérabilité (en suivant le scenario NM-S) et l'activité `exploit`, décrite dans la figure 4, modélise la disponibilité d'un *exploit* avant la publication de la vulnérabilité (en suivant le scenario M-S). Les préconditions, post-conditions et les paramètres caractérisant cette activité sont différents selon le scénario considéré. En effet, la publication de la

vulnérabilité augmente la probabilité que la population des attaquants crée un *exploit* puisqu'une plus grande proportion de cette population est au courant de l'existence de la vulnérabilité. L'existence ou la non-existence de l'*exploit* sont modélisées par les deux places respectivement nommées `E` pour "exploit") et `NE` (pour "no exploit").

5.2.2. *La modélisation des comportements des attaquants et de l'administrateur et des états du système*

La modélisation du comportement de l'administrateur est implicitement prise en compte dans la modélisation des différents états du système. Initialement, le système est dans l'état *ok*. L'activité `install` modélise l'installation du composant vulnérable. Le système passe donc dans l'état *vulnérable*. Il devient *exposé* dès qu'il existe un *exploit* (modélisé par la place `E`). Cet événement est modélisé par l'activité instantanée `test`. Les conditions pour la sensibilisation de cette activité sont définies dans la porte d'entrée associée à l'activité : l'existence de l'*exploit* et l'état vulnérable du système sont les conditions nécessaires pour que le système devienne *exposé* (modélisé par la place `Exposé`). L'utilisation de l'*exploit* par un attaquant sur le système est modélisé par trois activités `attaqueVd`, `attaqueVp`, `attaqueVc`, correspondant à une attaque perpétrée durant les différentes phases du cycle de vie de la vulnérabilité. Il est à noter que l'activité `attaqueVd` n'existe pas lorsque le scénario de découverte non malveillante est considéré (cf. figure 4). Lorsqu'une telle attaque est perpétrée avec succès, le système devient *compromis*. Il reste dans cet état jusqu'à ce que le correctif soit appliqué par l'administrateur, à partir de l'instant où celui-ci est disponible. Cette action est modélisée par l'activité `correctionC` : cela signifie que la vulnérabilité a été corrigée et ne peut plus être exploitée de nouveau. Cependant, le système n'est pas sûr tant que les dégâts causés par l'intrusion n'ont pas été complètement réparés. Cet état transitoire est modélisé par la place `Corrigé`. A partir de cet état, l'administrateur doit corriger et assainir le système, l'amenant dans l'état *sûr*. Il est à noter que le correctif de la vulnérabilité doit être appliqué dès que celui-ci est disponible, et tant que possible avant que le système ne soit victime d'un attaquant exploitant la vulnérabilité. Cette application du correctif peut avoir lieu dans deux situations distinctes : 1) le système est seulement *vulnérable* et il n'existe pas encore d'*exploit* disponible ; 2) le système est dans l'état *exposé* mais n'a pas encore été la cible d'une attaque. Dans les deux cas, le système devient *sûr*.

6. Caractérisation des événements du cycle de vie de la vulnérabilité

Pour quantifier les mesures que nous avons définies dans la section 4, il est important d'attribuer des paramètres réalistes aux activités de nos modèles. Cette section est dédiée à la caractérisation de certains de ces paramètres, basée sur des données réelles. A notre connaissance, les quelques études présentées dans la section 2 sont les rares existantes sur la caractérisation des événements du cycle de vie de la vulnérabilité que nous considérons dans notre approche et ces travaux ne peuvent pas être réutilisés. Ainsi, cette section présente notre travail de caractérisation visant à quantifier la probabilité d'occurrence des événements du cycle de vie de la vulnérabilité,

Base de données	NVD	Security Focus	OSVDB	Secunia
Date de découverte	non	non	oui	non
Date de publication	oui	oui	oui	oui
Date de publication du correctif	non	non	oui	non
Date d'apparition de l'exploit	non	non	oui	non

Tableau 1. *Comparaison des bases de données de vulnérabilités*

basé sur des données réelles. Dans un premier temps, nous présentons les bases de données existantes que nous avons examinées. Ensuite, nous présentons les données choisies pour la caractérisation des évènements du cycle de vie. Enfin, les deux dernières sections présentent la démarche et les résultats obtenus pour la caractérisation de chaque évènement du cycle de vie de la vulnérabilité. Des résultats préliminaires abordant cet aspect ont été présentés dans (Vache, 2009c).

6.1. *Les bases de données de vulnérabilités et les rapports statistiques existants*

Plusieurs organisations collectent et étudient les vulnérabilités. Certaines d'entre elles produisent régulièrement des rapports donnant des informations et les tendances sur l'évolution des vulnérabilités : par exemple, Symantec Corporation édite une étude bi-annuelle présentant les derniers types de vulnérabilités et les analyses quantitatives des données enregistrées grâce aux produits vendus par Symantec tels que leur antivirus (Symantec Enterprise Security, 2009). D'autres rapports existent, tels que le "X Force trends and risk report" (IBM, 2009b) qui classifie, par exemple, les systèmes d'exploitation en fonction du nombre de vulnérabilités publiées (IBM, 2009a).

Des données sont également disponibles dans plusieurs bases de données qui enregistrent les vulnérabilités et certaines de leurs caractéristiques. La base de données NVD (*National Vulnerability Database*) (NIST, 1999), gérée par le NIST (*National Institute of Standards and Technology of the United States*) et associée avec la référence CVE (*Common Vulnerabilities and Exposures*), archive les vulnérabilités depuis 1999 et fournit une évaluation de chaque vulnérabilité basée sur le CVSS (Common Vulnerability Scoring System) (Mell *et al.*, 2007). La base de données Security Focus (Security Focus, 2002) est gérée par Symantec Corporation depuis 2002 et contient plus de 35 000 vulnérabilités enregistrées depuis 1998. La base de données OSVDB (*Open Source Vulnerability DataBase*) a été créée en 2002 par la communauté de la conférence Black Hat. Elle contient plus de 52 000 vulnérabilités enregistrées depuis 1998 (OSF, 2002) et est disponible au grand public depuis 2004. Secunia, une organisation privée qui fournit des services pour la sécurité des systèmes et l'analyse de vulnérabilités possède également une base de données depuis 2002 (Secunia, 2002). La base de données indique elle-aussi la sévérité basée sur le système d'évaluation CVSS. Les caractéristiques de chaque base de données de vulnérabilités sont résumées dans le tableau 1, indiquant les événements pour lesquels la date est disponible.

Evénements	Découverte	(%)	Publication	(%)	Correction	(%)	Exploit	(%)
Découverte	3961	100.00	3926	7.68	148	12.86	2131	11.93
Publication	3926	99.12	51099	100.00	871	75.67	17857	100.00
Correction	148	3.74	871	1.71	1151	100.00	290	1.62
Exploit	2131	53.80	17857	34.95	290	25.20	17857	100.00

Tableau 2. *Informations fournies par l'ensemble de vulnérabilités*

6.2. *Caractérisation des événements*

Pour estimer les paramètres caractérisant l'occurrence des événements du cycle de vie de la vulnérabilité décrits dans nos modèles, il est nécessaire d'obtenir un ensemble de données aussi complet que possible. L'ensemble le plus exhaustif est, bien entendu, l'union des données fournies par les différentes bases de données. Malheureusement, les bases de données de vulnérabilités sont très hétérogènes. Même si CVE semble être une unique référence, elle n'est pas renseignée pour chaque vulnérabilité de chaque base de données. Il n'est donc pas aisé de fusionner et corréler les informations situées dans différentes bases de données. Aussi, avant d'analyser des données, nous avons dû choisir la base de données la plus pertinente pour notre étude. Notre objectif étant de caractériser les événements du cycle de vie de la vulnérabilité, nous nous intéressons en premier lieu aux dates de ces événements. La base de données OSVDB est celle satisfaisant le mieux nos exigences puisque c'est la seule enregistrant ce type d'information pour tous les événements. Cette base fournit un ensemble important de données : nous avons analysé 52 000 vulnérabilités extraites de la base de données et enregistrées depuis décembre 1998. Pour chaque vulnérabilité, l'identifiant OSVDB, les catégories de la vulnérablité, et les dates de découverte, publication de la vulnérabilité, publication du correctif et apparition de l'*exploit* sont enregistrées, lorsque ces informations sont disponibles, ce qui n'est malheureusement pas le cas pour chaque vulnérabilité. La section suivante présente les premières étapes de notre analyse.

6.3. *Analyses préliminaires*

Avant d'analyser l'ensemble de données pour déterminer quelles distributions de probabilité correspondent aux différents intervalles de temps, cette section décrit une analyse préliminaire de l'ensemble de données, basée sur les informations résumées dans le tableau 2.

Le nombre contenu dans la cellule(i, j) indique le nombre d'événements pour lesquels les dates d'occurrence des événements i et j sont disponibles. Le pourcentage associé indique la proportion de ce nombre de vulnérabilités relativement à celui indiqué dans la colonne. Considérons par exemple l'ensemble de vulnérabilités pour lesquelles les dates de découverte et de publication sont renseignées. Cet ensemble de 3926 vulnérabilités représente une petite proportion du nombre total de vulnérabilités (7,8%) mais représente aussi 99,12% de l'ensemble de vulnérabilités pour lesquelles

la date de découverte est disponible. Ce petit nombre de vulnérabilités peut s'expliquer par le fait que la découverte de la vulnérabilité n'est pas un événement officiellement publié.

Considérons maintenant l'ensemble des vulnérabilités pour lesquelles les dates de publication de la vulnérabilité et de publication du correctif sont disponibles. Il compte 871 vulnérabilités et représente seulement 1,71% de l'ensemble de vulnérabilités pour lesquelles la date de publication de la vulnérabilité est disponible, mais 75,67% des vulnérabilités pour lesquelles la date de publication du correctif est disponible. Deux raisons semblent pouvoir expliquer le faible nombre de vulnérabilités pour lesquelles la date de publication du correctif est disponible :

– Seule une faible proportion des vulnérabilités publiées ont un correctif disponible. Cette explication semble possible car nous considérons que la vulnérabilité peut être publiée par un autre organisme que le développeur du composant vulnérable (Jumratjaroenvanit *et al.*, 2008) ;

– Cet ensemble de données n'est pas exhaustif et le fait que la date de publication du correctif ne soit pas renseignée ne signifie pas qu'il n'existe pas de correctif.

Finalement, examinons l'ensemble de vulnérabilités pour lequel les dates d'apparition de l'*exploit* et la date de publication de la vulnérabilité sont renseignées : il représente seulement 34,34% de l'ensemble total étudié. Ceci met en évidence que l'apparition de l'*exploit*, comme la publication du correctif, n'est pas un événement systématique.

Il est important de prendre en compte cette information dans le paramétrage de nos modèles. Dans la section suivante, nous présentons les distributions de probabilités caractérisant les occurrences des événements du cycle de vie de la vulnérabilité, basées sur les données présentées dans le tableau 2.

6.4. *Analyse des données et caractérisation des événements*

Pour caractériser les probabilités d'occurrence d'un événement depuis un état *i* menant à un état *j*, nous sélectionnons les vulnérabilités pour lesquelles les dates *ti* et *tj* sont renseignées et évaluons la durée *tj-ti*. Un nouvel ensemble de données, composé des durées *tj-ti*, est obtenu. Nous avons besoin de classer ces données pour être capable de les analyser et de déterminer la distribution de probabilité la plus appropriée. Organiser les données en classes pour estimer la distribution de probabilité empirique nous permet de nous concentrer sur la tendance générale tout en minimisant l'impact des plus petites variations. Nous avons déterminé le nombre de classes grâce à la formule de Sturges (Sturges, 1926) qui est couramment utilisée en statistiques dans cet objectif (Saporta, 2009). Nous avons fait le choix que toutes les classes contiendraient le même nombre de données (Vache, 2009c).

Une fois les données classées, nous avons utilisé l'outil EasyFit (Mathwave, 2004) pour confronter les distributions de probabilités empiriques obtenues avec différentes

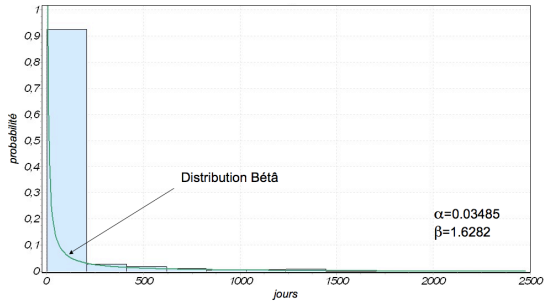

Figure 5. *Intervalles de temps entre dates de écouverte et de publication de la vulnérabilité*

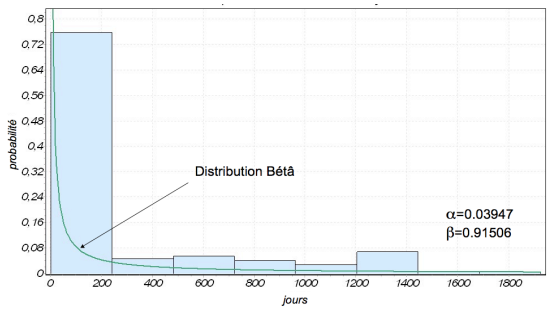

Figure 6. *Intervalles de temps entre dates de publication de la vulnérabilité et d'apparition de l'exploit*

distributions de probabilités. Nous utilisons le test statistique de Kolmogorov Smirnov pour vérifier l'adéquation entre distributions de probabilité empirique et théorique.

Nous analysons les données afin de caractériser les événements de publication de la vulnérabilité, publication du correctif et apparition de l'*exploit*. Comme l'ensemble des données que nous utilisons fournit la date de découverte de la vulnérabilité mais pas la date de déploiement du composant vulnérable, il n'est pas possible de caractériser la distribution associée à l'événement de découverte de la vulnérabilité.

6.4.1. *Caractérisation de l'événement de publication de la vulnérabilité*

La publication de la vulnérabilité peut survenir après la découverte non malvaillante de la vulnérabilité ou après l'utilisation de l'*exploit* par la population d'attaquants, comme détaillé dans la section 3.1.

Durée	Evénement	α	β	P-valeur
$t_p - t_d$	publication de la vulnérabilité (NM-S)	0.03485	1.6282	0.38
$t_c - t_p$	publication du correctif	0.00352	0.62362	0.41
$t_e - t_p$	apparition de l'*exploit* (NM-S)	0.00090	1.8666	0.35
$t_e - t_d$	apparition de l'*exploit* (M-S)	0.02916	1.5813	0.38
$t_d - t_e$	publication de la vulnérabilité (M-S)	0.03947	0.91506	0.34

Tableau 3. *Résumé des paramètres de la distribution Bêta*

Tout d'abord, nous étudions l'événement de publication de la vulnérabilité dans le contexte du scénario de découverte non malveillante (NM-S). Il y a 3926 vulnérabilités dans la base de données OSVDB dont les dates de découverte et de publication sont renseignées. 708 ont été découvertes et publiées le même jour, et 3218 ont été publiées 1 jour ou plus après leur découverte.

La figure 5 représente l'histogramme représentant la distribution empirique de l'intervalle de temps entre la découverte et la publication de la vulnérabilité. La première barre a une valeur de 92,51% et la seconde de 2,75%. Cette forte décroissance peut être décrite par une distribution Bêta. Cela est confirmé par le test de Kolmogorov Smirnov. Les paramètres et les p-values du test de Kolmogorov Smirnov sont résumées dans le tableau récapitulatif 3 : *td* représente la date de découverte de la vunérabilité, *tp* la date de publication de la vulnérabilité, *tc* la date de publication du correctif, et *te* la date d'apparition de l'*exploit*. Les paramètres α et β sont les paramètres de forme de la distribution Bêta, donc la densité de distribution est :

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$$

Considérons le second scénario dans lequel la vulnérabilité est découverte par une personne malveillante (M-S). Nous avons besoin de tenir compte de la date de publication de la vulnérabilité *tp* ainsi que la date d'apparition de l'*exploit te* telles que *tp-te* soit positif. Il y a 222 vulnérabilités correspondant à ce critère. L'intervalle de temps entre ces deux événements varie de 1 à 2151 jours. La distribution de probabilité correspondant à cet ensemble de données est une distribution Bêta. La distribution empirique des données et la distribution Bêta associée sont illustrées sur la figure 6.

6.4.2. *Caractérisation de l'événement de publication du correctif*

La publication du patch est étudiée suivant la même méthodologie que la publication de la vulnérabilité. Nous avons analysé un ensemble de 871 vulnérablités. Pour 712 d'entre elles, les dates de publication de la vulnérabilité et du correctif sont identiques. L'intervalle de temps varie entre 0 et 759 days. Il apparaît que la distribution Bêta est celle s'adaptant le mieux à notre distribution empirique en satisfaisant le test de Kolmogorov Smirnov. Les paramètres estimés pour cette distribution sont donnés dans le tableau 3.

6.4.3. *Caractérisation de l'apparition de l'exploit*

Pour caractériser la probabilité d'apparition de l'*exploit*, nous avons besoin de comparer le date d'apparition de l'exploit avec la date de découverte de la vulnérabilité (dans le cas du scénario M-S) et de publication de la vulnérabilité (dans le cas du scénario NM-S).

Tout d'abord, nous considérons l'ensemble de 2131 vulnérabilités pour lesquelles les dates d'apparition de l'*exploit* et de découverte de la vulnérabilité sont renseignées et 389 d'entre elles ont été découvertes et exploitées le même jour. Pour ces vulnérabilités, il est vraisemblable que l'origine de leur découverte soit maveillante. Nous avons observé que, dans ce cas aussi, cet ensemble de données peut être caractérisé par une loi Bêta en satisfaisant le test de Kolmogorov-Smirnov.

Examinons les vulnérabilités pour lesquelles les dates de publication de la vulnérabilité et d'apparition de l'*exploit* sont disponibles. Nous nous basons sur un ensemble de 17857 vulnérabilités. Notons que 1) pour 222 vulnérabilités, l'*exploit* apparaît avant que la vulnérabilité ne soit publiée (ces vulnérabilités sont utilisées pour la caractérisation de l'événement de publication de la vulnérabilité dans le scénario M-S) ; 2) pour 17077 vulnérabilités, l'*exploit* et la vulnérabilité sont publiés le même jour ; 3) pour 558 vulnérabilités, l'*exploit* apparaît après la publication de la vulnérabilité. Dans cette section, nous analysons les vulnérabilités appartenant aux deux derniers cas car elles correspondent au scénario NM-S. Il est important de noter que la plupart des vulnérabilités sont publiées et exploitées le même jour, mettant en évidence l'impact important de la publication de la vulnérabilité sur le reste du cycle de vie. En considérant ces deux ensembles de vulnérabilités comptant respectivement 17077 et 558 vulnérabilités, l'analyse de la distribution de l'intervalle de temps entre la publication de la vulnérablité et l'apparition de l'exploit a montré que la distribution Béta correspondait, et ce confirmé par le test de Kolmogorov-Smirnov.

7. Traitement des modèles et obtention des mesures quantitatives

Pour obtenir les valeurs des mesures présentées dans la section 4.2, l'étape suivante de notre approche est l'exécution des modèles que nous avons définis. Dans cette section, nous présentons d'abord comment nous avons paramétré les activités des modèles et nous analysons ensuite les résultats obtenus à partir de ces modèles. Les paramètres et leurs valeurs sont résumés dans le tableau 4. Nous supposons que la vulnérabilité est déjà installée dans le système. Par conséquent, nous ne considérons pas l'activité `install` et supposons que le système se trouve initialement dans l'état *vulnérable*.

Activité	Distribution	Paramètre	Valeur
`attaqueVp : NM-S`	exponentielle	taux	$0.5\,jours^{-1}$
`attackVc : NM-S`	exponentielle	taux	$0.1\,jours^{-1}$
`attaqueVd : M-S`	exponentielle	taux	$1\,jours^{-1}$
`attaqueVp : M-S`	exponentielle	taux	$5\,jours^{-1}$
`attaqueVc : M-S`	exponentielle	taux	$1\,jours^{-1}$
`correctionVul`	normale	moyenne	$1/30\,jours$
		variance	$0.5\,jour^{-2}$
`correctionExp`	normale	moyenne	$0.5/15\,jours$
		variance	$0.5\,jour^{-2}$
`correctionC`	normale	moyenne	$0.1/3\,jours$
		variance	$0.5\,jour^{-2}$
`reparation`	normale	moyenne	$3\,jours$
		variance	$0.5\,jour^{-2}$

Tableau 4. *Paramétrage des activités modélisant le comportement des attaquants et de l'administrateur dans les modèles NM-S et M-S*

7.1. *Description des paramètres*

7.1.1. *Le cycle de vie de la vulnérabilité*

Dans le scénario NM-S, l'analyse préliminaire a montré que la quasi totalité des vulnérabilités sont publiées. Nous appliquons donc la distribution Bêta avec les paramètres fournis par notre étude. Après la publication de la vulnérabilité, deux événements peuvent se produire : l'apparition de l'*exploit* et la publication du correctif.

D'après la base de données, seulement 2% des vulnérabilités publiées ont également un correctif publié. Cette valeur de 2% pour la publication d'un correctif semble très faible. Nous avons décidé d'exécuter le modèle en considérant cette valeur, fournie par l'analyse de la base, mais aussi en considérant plusieurs autres valeurs pour la probabilité d'existence d'un correctif, notée *p*, (5%, 10%, 50% and 100%) pour des analyses de sensibilité. Lorsque le correctif existe, la probabilité de publication est modélisée par une distribution Bêta.

L'analyse de la base de données de vulnérabilités montre que seulement 34.5% des vulnérabilités ont un exploit associé. Dans ce cas, la durée entre la publication de la vulnérabilité et l'apparition de l'exploit peut être décrite par une distribution Bêta avec les paramètres décrits dans le tableau 3.

Dans le scénario de découverte malveillante, la découverte de la vulnérabilité amène à l'apparition de l'exploit. Cet événement est modélisé par une distribution Bêta avec les paramètres déterminés par l'analyse de la base de données. L'intervalle de temps entre apparition de l'exploit et publication de la vulnérabilité est également modélisé par une distribution Bêta. Dans ce scénario, nous faisons l'hypothèse que la publication du correctif est un événement qui se produit inévitablement. Ce choix est justifié par le fait qu'une vulnérabilité découverte par une personne malveillante, et donc exploitée avant sa publication, représente une menace très sérieuse. C'est la raison pour laquelle nous supposons que le correctif sera publié avec une probabilité

égale à 1 dans ce cas de figure. L'intervalle de temps entre la publication de la vulnérabilité et la publication du correctif est également modélisé par une distribution Bêta.

7.1.2. *Processus d'attaque - exploitation de la vulnérabilité*

Le processus d'attaque est différent selon le scénario de découverte considéré. La vulnérabilité peut être exploitée avant sa publication seulement dans le scénario M-S et est modélisé par l'activité `attaqueVd`. Les deux autres activités modélisant la phénomène d'attaque, `attaqueVp` et `attaqueVc`, sont présentes dans les deux modèles. Ces trois activités sont décrites avec des distributions exponentielles, comme dans (Ortalo *et al.*, 1999; Kuhl *et al.*, 2007). Nous considérons différents taux selon la phase du cycle de vie de la vulnérabilité considérée. Les taux d'attaque sont plus grands pour le scénario M-S car la vulnérabilité représente, dans ce cas de figure, une plus grande menace pour le système. De plus, le taux d'attaque lorsque la vulnérabilité est publiée est plus grand qu'à n'importe quel autre moment du cycle de vie, puisque tous les attaquants peuvent potentiellement perpétrer une attaque avec succès.

7.1.3. *Application du correctif et réparation du système*

Le correctif peut être appliqué par l'administrateur en plusieurs circonstances : lorsque le système est *vulnérable*, *exposé* ou *compromis*. L'application du correctif est donc modelisée différemment selon ces circonstances, et respectivement par les activités `correctionVul`, `correctionExp`, `correctionC`. Nous supposons que ces activités peuvent être décrites par des distributions normales. Il n'existe pas de travaux sur le comportement de l'administrateur, cependant, nous choisissons cette distribution pour traduire une latence entre l'événement et la réactivité de l'administrateur. Plus grande est la menace pour le système (en considérant les états *vulnérable*, *exposé* et *compromis*) plus petit est le temps moyen entre la publication du correctif et l'application de ce dernier. L'application du correctif empêche le système d'être la cible d'autres attaques exploitant la vulnérabilité. Cependant, cette action n'est pas suffisante pour rendre le système *sûr*. La réparation du système correspond àu nettoyage et à la récupération du système, étape nécessaire pour sécuriser le système. Dans l'exécution de nos modèles, nous considérons qu'il faut une journée pour réparer le système. Dans notre étude, nous analysons deux comportements d'administrateurs différents : un laxiste et un rigoureux. L'administrateur laxiste met son système à jour une fois par mois quand l'administrateur rigoureux le met à jour tous les jours.

7.2. *Résultats*

7.2.1. *Scénario de découverte non malveillante (NM-S)*

Cette section présente les résultats obtenus lors de l'exécution du modèle NM-S. Nous nous concentrons sur les états du système nécessaires à la quantification des mesures. La première partie de cette section est dédiée à l'étude de la probabilité pour le système d'être dans un des états *vulnérable*, *exposé*, ou *corrigé* avant de s'inté-

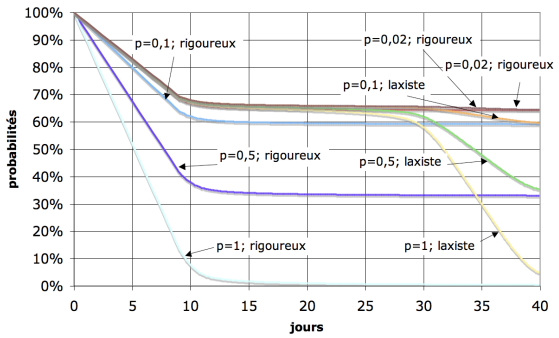

Figure 7. *Evolution de la probabilité pour le système d'être dans l'état vulnérable dans le scénario NM-S*

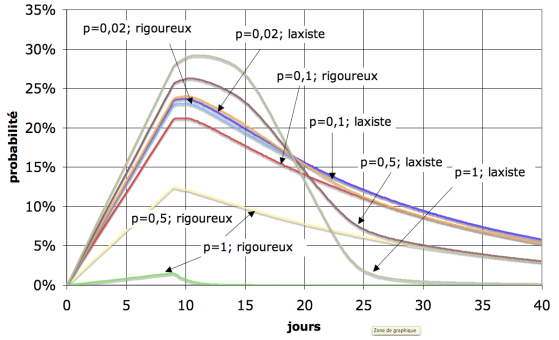

Figure 8. *Evolution de la probabilité pour le système d'être dans l'état exposé dans le scénario NM-S*

resser à l'obtention des mesures présentées dans la section 4.2. Les deux comportements des administrateurs (rigoureux et laxiste) sont modélisés par les trois activités `correctionVul`, `correctionExp` and `correctionC`.

La figure 7 montre l'évolution de la probabilité pour le système d'être dans l'état *vulnérable*. Elle met en évidence l'influence de l'existence d'un correctif (probabilité p) mais également la différence entre les deux comportements de l'administrateur. Lorsque l'administrateur est rigoureux, la probabilité pour le système d'être dans l'état *vulnérable* décroît rapidement à cause de la disponibilité de l'exploit (qui amène le système dans l'état *exposé*) mais aussi grâce au fait que l'administrateur applique le correctif dès que celui-ci est publié. Les deux événements ont lieu au bout d'1 jour en moyenne après la publication de la vulnérabilité. Dans la cas d'un administrateur laxiste, la probabilité décroit à cause de la disponibilité de l'exploit (après 1 jour en

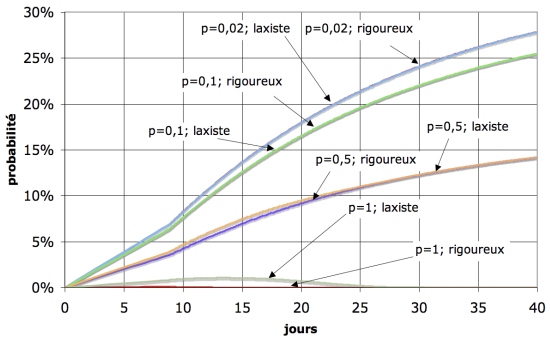

Figure 9. *Evolution de PCNR(t) dans le scénario NM-S*

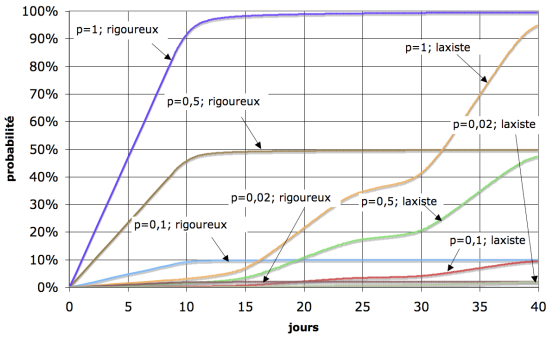

Figure 10. *Evolution de PS(t) dans le scénario NM-S*

moyenne) qui amène le système dans l'état *exposé*. Cependant, il faut beaucoup plus de temps à l'administrateur pour appliquer le correctif, comme l'illustre la lente décroissance de la courbe (commençant en moyenne au jour 30).

Les courbes de la figure 8 montrent l'évolution de la probabilité d'avoir le système *exposé*. La courbe commence par croître à cause de la disponibilité de l'exploit. Cette croissance est cependant moins importante pour l'administrateur rigoureux qui a déjà appliqué le correctif avant. La décroissance peut être causée par deux événements : l'application du correctif ou une attaque perpétrée avec succès. Dans le cas considérant une faible probabilité d'existence du correctif (2% et 10%), le processus d'attaque est l'événement ayant le plus d'impact, ayant un taux d'occurrence plus élevé. A l'inverse, pour les autres cas (50% et 100%), la décroissance de la courbe est due principalement à l'application du correctif et non à l'exploitation de la vulnérabilité.

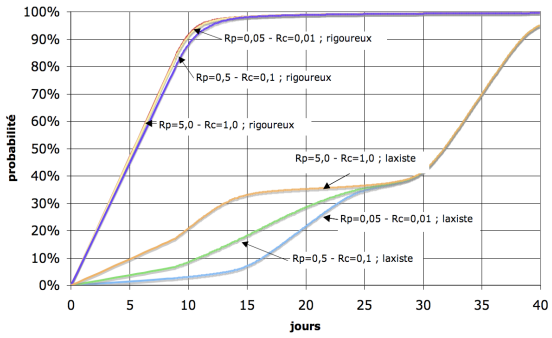

Figure 11. *Evolution de PS(t) dans le scénario NM-S considérant différents taux d'attaque*

L'évolution de la mesure *PCNR(t)* (cf. figure 9) est très similaire à l'évolution de la mesure *PPC(t)*. En considérant une faible probabilité d'existence du correctif (2% et 10%), les courbes correspondant à l'administrateur laxiste et rigoureux ne peuvent pas être distinguées et croient rapidement jusqu'à la valeur 34,5%, qui est la probablité maximale d'existence d'un exploit (cf. section 6). Lorsque l'on considère une probabilité d'existence du correctif égale à 100%, il est important de constater la différence entre leurs courbes obtenues, mettant en lumière les comportements des deux administrateurs : même si l'administrateur laxiste a une valeur *PCNR(t)* faible, celle-ci reste supérieure à la valeur de l'administrateur rigoureux. De plus, la différence entre les courbes est due au fait que la probabilité d'atteindre l'état *corrigé*, sachant que le système a été compromis, est plus faible dans le cas de l'administrateur rigoureux car le système a une probabilité plus grande d'avoir été corrigé avant qu'une attaque avec succès n'ait été perpétrée.

La probabilité pour le système d'être dans l'état *corrigé* a un impact direct sur la probabilité que le système atteigne l'état *sûr*, illustré dans la figure 10. La mesure *PS(t)* dépasse (dans le cas d'un administrateur rigoureux) 90% au jour 10 dans le cas d'une probabilité d'existence d'un correctif de 100%, comparé à 9% lorsque la probabilité d'existence du correctif est de 10% seulement. La figure 10 met également en évidence la différence entre les comportements des deux administrateurs : en considérant la même probabilité d'existence du correctif, il faut au moins 25 jours supplémentaires à l'administrateur laxiste pour atteindre une valeur similaire de la mesure *PS(t)*, comparé à celle de l'administrateur rigoureux, et ce certainement après que le système ait été compromis et réparé.

Pour finir, nous faisons varier les taux d'attaque pour évaluer leur impact : *Rp* et *Rc* sont les probabilités associées aux activités `attackVp` et `attackVc`. La figure 11 montre l'évolution de la probabilité *PS(t)* en considérant que le correctif existe dans tous les cas. L'évolution de la probabilité considérant plusieurs taux d'attaque est très

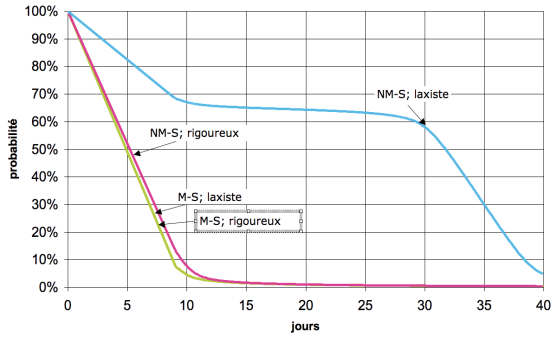

Figure 12. *Evolution de la probabilité pour le système d'être dans l'état vulnérable dans le scénario M-S comparé au scénario NM-S*

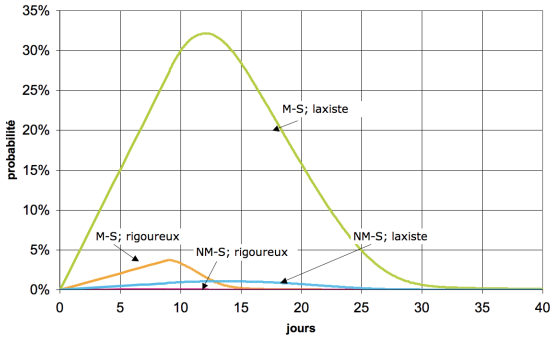

Figure 13. *Evolution de PCNR(t) dans le scénario M-S comparé au scénario NM-S*

similaire lorsque l'on considère un administrateur rigoureux, car celui-ci applique le correctif dès que ce dernier est publié, laissant peu de temps aux attaquants d'exploiter la vulnérabilité. Dans le cas où l'on considère un administrateur laxiste, il y a une plus grande sensibilité à la variation du taux d'attaque. Si nous considérons un taux d'attaque important, le système devient sûr plus rapidement. Cependant, cette application rapide du correctif est due au fait que le système a été compromis puis réparé avant de devenir sûr.

7.2.2. *Scénario de découverte malveillante (M-S)*

Comme cela a été fait pour le scénario NM-S, nous présentons la probabilité pour le système d'être dans un des différents états et en déduisons les valeurs pour les mesures définies dans la section 4. Pour chaque état ou mesure considérés, nous com-

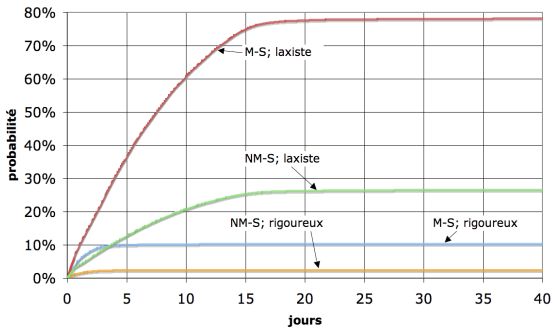

Figure 14. *Evolution de PC(t) dans le scénario M-S comparé au scénario NM-S*

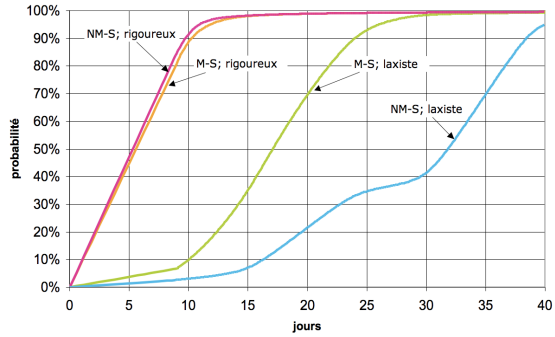

Figure 15. *Evolution de PS(t) dans le scénario M-S comparé au scénario NM-S*

parons les tendances pour les deux comportements d'administrateur mais également pour les deux scénarios. Tout d'abord, la figure 12 représente l'évolution pour le système d'être dans l'état vulnérable considérant qu'il existe toujours un correctif (p=1). Les deux courbes correspondant aux comportements des deux administrateurs dans le scénario M-S sont très similaires et décroissent plus rapidement que dans le cas du scénario NM-S. Cela est dû au fait que l'exploit est disponible avant le correctif : même si l'administrateur est rigoureux, le correctif ne peut pas être appliqué et amène le système dans l'état *exposé*. La figure 13 illustre l'évolution de la mesure *PCNR(t)* dans le scénario M-S, en comparaison avec le scénario NM-S. La différence entre les deux comportements d'administrateur est importante : la mesure *PCNR(t)* a une valeur maximale de 32% avec un administrateur laxiste et 4% pour un administrateur rigoureux. La mesure *PCNR(t)* est intéressante car elle quantifie la probabilité pour le système d'être en danger. La figure 14 montre l'évolution de la mesure de *PC(t)*. Cette mesure permet de déterminer si le système est ou a été *en danger* en considé-

rant la vulnérabilité. En considérant le même comportement d'administrateur, *PC(t)* atteint des valeurs plus faibles dans le cas du scénario NM-S, puisque l'administrateur peut appliquer le correctif avant que l'exploit ne soit disponible. La différence entre les deux comportements d'administrateur est également mise en évidence puisque la probabilité d'avoir le système compromis atteint 60,6% au jour 10 dans le cas d'un administrateur laxiste et seulement 10,2% dans le cas d'un administrateur rigoureux, dans le scénario M-S. Enfin, la figure 15 illustre l'évolution de la mesure *PS(t)*, qui mesure la probabilité pour le système d'être dans l'état *sûr*, même si celui-ci a été compromis. L'état *sûr* est un état absorbant dans le modèle, les courbes ne cessent donc de croître pour atteindre la valeur 1. Il est intéressant de noter que les deux courbes correspondant à un administrateur rigoureux suivent la même tendance. Dans le cas de l'administrateur laxiste, le correctif est appliqué plus tôt dans le scénario M-S à cause de la valeur plus élevée de la probabilité d'attaque perpétrée avec succès. La mesure *PS(t)* est intéressante puisqu'elle peut être utilisée pour quantifier le temps nécessaire permettant de garantir la probabilité minimale (déterminée par l'administrateur) pour le système pour être *sûr*.

7.2.3. *Etude d'un administrateur au comportement intermédiaire*

Dans cette étude, nous avons considéré deux comportements extrêmes : un administrateur rigoureux et un administrateur laxiste. Cependant, la réactivité de la majorité des administrateurs va se situer entre ces deux comportements. Pour conclure cette étude, nous donnons les résultats relatifs à un administrateur dont le comportement est intermédiaire entre ces deux cas extrêmes, que nous qualifions de moyen. Les temps de réaction de cet administrateur suivent les même rapports que ceux déja modélisés. Les temps moyens de réaction de cet administrateur sont :

– 7 jours lorsque le système est *vulnérable* ;

– 3.5 jours lorsque le système est *exposé* ;

– 0.7 jours lorsque le système est *compromis*.

Comme pour la comparaison entre les deux scénarios M-S et NM-S, nous nous concentrons sur la mesure *PS(t)*. Les tableaux 5 et 6 montrent l'évolution de la mesure *PS(t)* en considérant les trois profils d'administrateurs et les deux scénarios.

Cet administrateur moyen a un comportement qui reste cohérent par rapport aux deux autres administrateurs. En effet, dans le scénario de découverte non malveillante, au bout de 10 jours, l'administrateur moyen applique le correctif avec une probabilité de 45.36%. Cette valeur montre une réactivité bien supérieure à celle de l'administrateur laxiste mais tout de même inférieure à celle de l'administrateur rigoureux. Les instants suivants montrent une évolution bien plus importante chez l'administrateur moyen, en comparaison de l'administrateur laxiste. Cela est dû à une plus grande réactivité de celui-ci en cas de danger lorsque le système est dans l'état *exposé* ou *compromis*. Au 20ème jour, l'administrateur moyen a une mesure *PS(t)* très proche de celle de l'administrateur rigoureux. Cependant, cela est dû en plus grande partie à l'application du correctif lors d'une compromission.

Instant	Rigoureux	Moyen	Laxiste
10 jours	0.9161	0.4536	0.0309
15 jours	0.9825	0.8696	0.0705
20 jours	0.9893	0.9800	0.2158
25 jours	0.9922	0.9891	0.3469

Tableau 5. *Evolution de la mesure PS(t) pour trois administrateurs dans le scénario NM-S*

Instant	Rigoureux	Moyen	Laxiste
10 jours	0.8866	0.6092	0.0995
15 jours	0.9808	0.9507	0.3512
20 jours	0.9891	0.9855	0.6961
25 jours	0.9922	0.9907	0.9297

Tableau 6. *Evolution de la mesure PS(t) pour trois administrateurs dans le scénario M-S*

L'évolution constatée dans le scénario de découverte malveillante est également cohérente. En effet, on peut constater que la probabilité *PS(t)* augmente plus rapidement que dans le scénario NM-S, comme c'est le cas pour les deux autres administrateurs. Cela est dû à la probabilité plus importante d'une attaque perpétrée avec succès mais aussi de l'impossibilité d'appliquer le correctif avant l'existence d'un exploit.

Quelque soit le scénario considéré, cette évolution est donc logique et confirme les tendances données pour les administrateurs laxiste et rigoureux.

8. Conclusion et perspectives

Dans cet article, nous avons présenté une approche de modélisation pour l'évaluation quantitative de la sécurité. Notre objectif était d'élaborer un processus d'évaluation pouvant fournir des mesures quantifiant les risques pour le système d'être compromis par l'exploitation d'une vulnérabilité. Notre étude se concentre sur la caractérisation et la modélisation du processus d'exploitation de la vulnérabilité et son impact sur l'état du système. Le première étape de cette approche est l'identification et la caractérisation des facteurs externes qui peuvent avoir un impact sur le processus d'exploitation de la vulnérabilité. Cette étude met en évidence trois facteurs ayant un impact sur la sécurité : le cycle de vie de la vulnérabilité et deux facteurs environnementaux, le comportement des attaquants et le comportement de l'administrateur. Les dépendances entre ces facteurs nous ont amenés à distinguer deux scénarios basés sur l'origine de la découverte de la vulnérabilité, malveillante ou non malveillante. En prenant en considération ces trois facteurs, nous avons identifié les différents états du système et défini quatre mesures probabilistes. Pour les évaluer, nous avons défini deux modèles SAN (un modèle par scénario) et associé des distributions de probabilités aux différentes activités des modèles. Les événements du cycle de vie de la vulnérabilité ont été caractérisés en se basant sur des données réelles contenues dans la base de donnés de vulnérabilités OSVDB. Les autres événements ont été caractérisés en se basant sur la littérature. Les traitements de ces modèles nous ont permis

d'obtenir les mesures définies. Cet ensemble de quatre mesures de la sécurité permet d'avoir une vision globale des risques pour le système en considérant ces trois facteurs externes. Ces mesures répondent aux questions : "quelle est la probabilité pour mon système d'être sûr en considérant ce type de vulnérabilité ?" ou "quelle est la probabilité pour mon système d'être compromis par une attaque perpétrée avec succès ?". Ces mesures peuvent être évaluées en considérant les facteurs externes que nous avons identifiés.

L'exécution des deux modèles permet l'obtention des mesures que nous avons définies, en considérant les deux scénarios basés sur la découverte, malveillante ou non malveillante, de la vulnérabilité. Cependant, il est intéressant de considérer le fait que la base de données contient une grande quantité de vulnérabilités correspondant au scénario NM-S. Les mesures évaluées en considérant ce scénario sont donc très importantes puisqu'elles reflètent le cas le plus fréquent.

Les résultats présentés dans cet article confirment des tendances similaires aux résultats présentés dans (McQueen *et al.*, 2006). Cependant, il est difficile de comparer tous les résultats car les auteurs se concentrent sur un attaquant particulier, avec ses compétences propres, plus que sur la sécurité du système en considérant un environnement plus complexe.

Cette approche présente plusieurs perspectives. Tout d'abord, il est important de procéder à de plus amples analyses de sensibilité pour mettre en évidence les impacts des événements du cycle de vie de la vulnérabilité. Puis, comme nous l'avons fait pour les événements du cycle de vie, il est intéressant de chercher à caractériser le processus d'attaque en se basant sur des données réelles. La principale difficulté est de trouver de telles données. Vous avons rencontré cette difficulté lors de la caractérisation des évenements du cycle de vie. Une de nos perspectives est de fournir une base de données de vulnérabilités la plus exhaustive possible, en se basant sur les bases de données disponibles. Mais comme nous l'avons évoqué dans ce papier, ceci représente un travail important de par l'hétérogénéité des bases existantes.

Ensuite, nous avons fourni des modèles avec des valeurs calculées depuis un ensemble global de vulnérabilités sans distinguer certaines caractéristiques des systèmes affectés par ces vulnérabilités. Pour être capable de fournir de mesures plus précises, il serait intéressant de classer les vulnérabilités selon le système d'exploitation, la sévérité de l'impact de la vulnérabilité sur les attributs de la sécurité ou encore l'accès nécéssaire pour exploiter la vulnérabilité, pour attribuer des valeurs encore plus précises aux activités du modèle caractérisant le cycle de vie de la vulnérabilité. Ce type d'information est disponible dans la base de données OSVDB. Les premières analyses menant à ces résultats sont actuellement en cours.

Enfin, nous envisageons aussi d'étendre nos modèles pour la prise en compte de plusieurs vulnérabilités, et des interdépendances que celles-ci peuvent avoir durant une attaque. En effet, une attaque peut comporter plusieurs étapes, chacune consistant en l'exploitation d'une vulnérabilité, comme cela est décrit par exemple dans (Dacier *et al.*, 1996). Le formalisme des SAN nous permet de concevoir de tels modèles. Les

premières reflexions concernant ces modèles à plusieurs vulnérabilités sont décrits dans (Vache, 2009b).

Ces modèles complexes permettraient de fournir les mêmes mesures quantitatives de la sécurité définies dans cet article, permettant d'integrer cette méthode d'évaluation quantitative dans une methodologie visant à améliorer les propriétés des systèmes vis-à-vis des malveillances, telle que celle proposée dans (Mead *et al.*, 2000). En effet, cette méthodologie décrit un processus itératif visant à améliorer la survivabilité d'un système en effectuant les changements nécessaires sur la configuration et l'architecture. Notre approche pourrait intervenir dans la phase d'analyse et d'évaluation d'une telle méthodologie.

9. Bibliographie

Alata E., Nicomette V., Kaaniche M., Dacier M., Herrb M., « Lessons learned from the deployment of a high-interaction honeypot », *EDCC '06 : Proceedings of the Sixth European Dependable Computing Conference*, IEEE Computer Society, p. 39-46, 2006.

Arbaugh W., Fithen W., McHugh J., « Windows of vulnerability : a case study analysis », *Computer*, vol. 33, n° 12, p. 52-59, Dec, 2000.

Arora A., Krishnan R., Telang R., Yang Y., « Impact of Vulnerability Disclosure and Patch Availability - An Empirical Analysis », *In Third Workshop on the Economics of Information Security*, 2004.

Balzarotti D., Monga M., Sicari S., « Assessing the risk of using vulnerable components », *Quality of Protection*, Springer US, p. 65-77, 2006.

CERIAS, The development of meaningful hacker taxonomy : a two dimensional approach, Technical Report n° 2005-43, CERIAS, 2005.

Dacier M., Vers une évaluation quantitative de la sécurité informatique, PhD thesis, Institut National Polytechnique, Toulouse, 1994.

Dacier M., Deswarte Y., Kaâniche M., « Information systems security », Chapman & Hall, Ltd., London, UK, UK, chapter Models and tools for quantitative assessment of operational security, p. 177-186, 1996.

Daly D., Deavours D. D., Doyle J. M., Webster P. G., Sanders W., « Möbius : An extensible tool for perfomance and dependability modeling », Schaumnurg, IL B.R. Haverkort, H. C. Bohnenkamp, and C. U. Smith (Eds.), p. 332-336, 2000.

European Communities, « Information Technology Security Evaluation Criteria », , , 1991.

Frei S., Security Econometrics - The Dynamics of (In)Security, Eth zurich, ph.d. dissertation, ETH Zurich, 2009.

Frei S., May M., Fiedler U., Plattner B., « Large-scale vulnerability analysis », *LSAD '06 : Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense*, p. 131-138, 2006.

IBM, « IBM Internet Security Systems X-Force 2008 Trend & Risk Report », , , 2009a.

IBM, « X-Force Trends Reports », , `http://www-935.ibm.com/services/us/iss/xforce/trendreports/`, 2009b.

ISO/CEI 15408, « Common Criteria for Information Technology Security Evaluation », , , 1996.

ISO/IEC 27001, « Requirements for information security management systems », , , 2005.

ISO/IEC 27002, « Code of practice for information security management », , , 2005.

Jha S., Sheyner O., Wing J., « Two formal analyses of attack graphs », *Proceedings. of the 15th IEEEComputer Security Foundations Workshop*, p. 49-63, 2002.

Jonsson E., Olovsson T., « A quantitative model of the security intrusion process based on attacker behavior », *IEEE Transactions on Software Engineering*, vol. 23, n° 4, p. 235-245, Apr, 1997.

Jumratjaroenvanit A., Teng-amnuay Y., « Probability of Attack Based on System Vulnerability Life Cycle », *ISECS '08 : Proceedings of the 2008 International Symposium on Electronic Commerce and Security*, IEEE Computer Society, Washington, DC, USA, p. 531-535, 2008.

Kuhl M. E., Kistner J., Costantini K., Sudit M., « Cyber attack modeling and simulation for network security analysis », *WSC '07 : Proceedings of the 39th conference on Winter simulation*, IEEE Press, Piscataway, NJ, USA, p. 1180-1188, 2007.

LeMay E., Ford M. D., Keefe K., Sanders W. H., Muehrcke C., « Model-based Security Metrics using ADversary view Security Evaluation (ADVISE) », *Proceedings of the 8th International Conference on Quantitative Evaluation of Systems (Qest 2011)*, Aachen, Germany, p. 191-200, 2011.

Mathwave, « The EasyFit tool », , `http://www.mathwave.com`, 2004.

McQueen M., Boyer W., Flynn M., Beitel G., « Time-To-Compromise Model for Cyber Risk Reduction Estimation », *Quality of Protection*, Springer US, p. 49-64, 2006.

Mead N., Ellison R., Linger R., Longstaff T., McHugh J., Survivable Network Analysis Method, Technical Report n° CMU/SEI-2000-TR-013, ESC-2000-TR-13, Carnegie Mellon, 2000. `http://www.cert.org/archive/pdf/00tr013.pdf`.

Mell P., Scarfone K., Romanosky S., « A Complete Guide to the Common Vulnerability Scoring System Version 2.0 », , `http://www.first.org/cvss/cvss-guide.html`, 2007.

Nicol D., Sanders W., Trivedi K., « Model-based evaluation : from dependability to security », *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, n° 1, p. 48 - 65, january, 2004.

NIST, « National Vulnerability Database », , `http://nvd.nist.gov`, 1999.

Ortalo R., Deswarte Y., Kaâniche M., « Experimenting with Quantitative Evaluation Tools for Monitoring Operational Security », *IEEE Transactions on Software Engineering*, vol. 25, p. 633-650, 1999.

OSF, « Open Source Vulnerability Database », , `http://osvdb.org`, 2002.

Ozment A., Schechter, Stuart E., « Milk or wine : does software security improve with age ? », *USENIX-SS'06 : Proceedings of the 15th conference on USENIX Security Symposium*, USENIX Association, Berkeley, CA, USA, 2006.

Rescorla E., « Is finding security holes a good idea ? », *IEEE Security & Privacy*, vol. 3, n° 1, p. 14-19, Jan.-Feb., 2005.

Rogers M. K., « A two-dimensional circumplex approach to the development of a hacker taxonomy », *Digital Investigation*, vol. 3, n° 2, p. 97 - 102, 2006.

Sanders W. H., Meyer J. F., *Stochastic activity networks : formal definitions and concepts*, Springer-Verlag New York, Inc., New York, NY, USA, p. 315-343, 2002.

Saporta G., *Probabilités, analyse de données et statistique*, Technip, 2ème édition, 2009.

Secunia, « Secunia Vulnerability Database », , `http://secunia.com/`, 2002.

Security Focus, « Security Focus Vulnerability Database », , `http://www.securityfocus.com`, 2002.

Sheyner O., Scenario Graphs and Attack Graphs, PhD thesis, Carnegie Mallon University, Pittsburgh, PA, 2004.

Sturges H. A., « The Choice of a Class Interval », *Journal of the American Statistical Association*, vol. 21, n° 153, p. 65-66, mar, 1926.

Swiler L., Phillips C., Ellis D., Chakerian S., « Computer-attack graph generation tool », *Proceedings of DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01*, vol. 2, p. 307-321, 2001.

Symantec Enterprise Security, « Symantec Global Internet Security Threat Report Ð Trends for 2008 », , , 2009.

U.S. Department Of Defense, « Trusted Computer Security Evaluation Criteria », , , 1985.

Vache G., « Environment Characterization and System Modeling Approach for the Quantitative Evaluation of Security », *Proceedings of the 28th International Conference on Computer Safety, Reliability and Security*, p. 89-102, 2009a.

Vache G., Evaluation quantitative de la sécurité informatique : approche par les vulnérabilités, PhD thesis, Institut National des Sciences Appliquées de Toulouse (INSA), Décembre, 2009b.

Vache G., « Vulnerability analysis for a quantitative security evaluation », *Proceedings of the International Symposium on Empirical Software Engineering and Measurement*, IEEE Computer Society, p. 526-534, 2009c.

Zuse H., *Software complexity : measures and methods*, Walter de Gruyter & Co., Hawthorne, NJ, USA, 1991.