Évaluation quantitative de la sécurité des systèmes d'information

Rodolphe Ortalo

N° d'ordre : 1418
Année 1998

THÈSE

présentée au

Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique

en vue d'obtenir le titre de

Docteur de l'Institut National Polytechnique de Toulouse

Spécialité : Informatique

par

Rodolphe ORTALO

Ingénieur Supélec

ÉVALUATION QUANTITATIVE DE LA SÉCURITÉ DES SYSTÈMES D'INFORMATION

Soutenue le 19 mai 1998 devant le jury :

Président	David POWELL
Rapporteurs	Jean-Jacques QUISQUATER
	Pierre ROLIN
Examinateurs	Frédéric CUPPENS
	Yves DESWARTE
	Laurent DUBIEF
	Claude NODOT
	Brian RANDELL

Rapport LAAS N° 98164

LAAS - CNRS
7, avenue du Colonel Roche 31077 Toulouse Cedex 4

Thèse de doctorat de Rodolphe Ortalo

“Évaluation quantitative de la sécurité des systèmes d’information”

Résumé

Cette thèse présente une méthode générale de spécification et d’évaluation quantitative de la sécurité des systèmes d’information. Cette méthode permet de surveiller les évolutions d’un système d’information pendant sa vie opérationnelle, ainsi que de comparer l’impact sur la sécurité de modifications éventuelles du fonctionnement. Elle s’appuie sur une spécification formelle de la politique de sécurité, complétée par un modèle des vulnérabilités du système réel en fonctionnement. Une mesure de la sécurité correspond alors à la difficulté pour un attaquant potentiel d’exploiter les vulnérabilités pour mettre en défaut les objectifs définis par la politique de sécurité.

La spécification de la politique de sécurité d’un système d’information nécessite la définition d’un cadre rigoureux, expressif, et suffisamment général pour être utilisable dans le contexte d’une organisation. La méthode définie et utilisée dans ce mémoire est basée sur une extension de la logique déontique, complétée par une représentation graphique.

Les vulnérabilités du système d’information sont représentées par un modèle appelé un graphe des privilèges. Ces vulnérabilités, qui doivent être recherchées dans le système, peuvent avoir des origines variées, comme la mauvaise utilisation des mécanismes de protection dans un système informatique, ou les délégations de pouvoirs dans une organisation. L’attribution d’un poids à ces vulnérabilités élémentaires permet de définir des mesures quantitatives de la sécurité globales et pertinentes.

La démarche est illustrée par deux exemples d’application pratique : l’étude du fonctionnement d’une agence bancaire de taille moyenne ; et l’observation de l’évolution de la sécurité d’un système informatique de grande taille en exploitation.

Mots-Clefs : sûreté de fonctionnement, sécurité, politique de sécurité, évaluation quantitative, système d’information, graphe des privilèges.

“Quantitative Evaluation of Information Systems Security”

Abstract

This dissertation presents a general method for the specification and quantitative evaluation of information systems security. This method allows to monitor the evolutions of an information system in operation, as well as to compare the impact on security of possible modifications of the functioning. It relies on a formal specification of the system security policy, augmented by a model of the vulnerabilities observed in the real system in operation. Then, a security measure represents the difficulty for an attacker to exploit the vulnerabilities and defeat the objectives defined in the security policy.

Information systems security policy specification necessitates the definition of a rigorous and expressive framework. Furthermore, the language should be general enough to be usable in the context of an organization. The method defined and used in this work is based on an extension of deontic logic, enriched with a graphical representation.

Vulnerabilities of the information system are described by a model called a privilege graph. These vulnerabilities, probed in the system, may have various origins, such as incorrect operation of the protection mechanisms of a computer system, or delegation of privileges in an organization. The assessment of a weight to these individual vulnerabilities allows the definition of highly relevant and global quantitative measures of security.

Two practical examples are presented to illustrate the methodology : the study of a medium-size bank agency ; and the observation of the security evolutions of a large computer system in operation.

Keywords : dependability, security, security policy, quantitative evaluation, information system, privilege graph.

À mon grand-père.

Avant-Propos

Les travaux présentés dans ce mémoire ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) du Centre National de la Recherche Scientifique (CNRS). Je tiens à remercier tout d'abord Alain Costes, ainsi que Jean-Claude Laprie, directeurs successifs du LAAS pendant mon séjour, de m'avoir accueilli au sein de ce laboratoire.

Je voudrais également exprimer ma reconnaissance à Jean-Claude Laprie, responsable du groupe de recherche "Tolérance aux fautes et Sûreté de Fonctionnement informatique" (TSF) à mon arrivée, pour m'avoir permis de mener mes travaux au sein de ce groupe; et à David Powell qui lui a succédé, pour m'avoir soutenu jusqu'à leur aboutissement.

Je tiens à remercier tout particulièrement Yves Deswarte pour m'avoir encadré tout au long de cette thèse. Il a été soutenu par Mohamed Kaâniche dans cette tâche. Leur rôle à tous deux a été primordial dans ce travail. Il m'est d'ailleurs tout aussi difficile de leur exprimer ma gratitude que de les dissocier dans ces remerciements. J'ai été très heureux de travailler avec eux.

Je remercie David Powell, Directeur de Recherche CNRS, pour l'honneur qu'il me fait en présidant mon jury de thèse, ainsi que:

- Jean-Jacques Quisquater, Professeur à l'Université Catholique de Louvain-la-Neuve;
- Pierre Rolin, Chef du Service Évaluation Veille et Compétence de la direction scientifique du Centre National d'Études des Télécommunications (CNET);
- Frédéric Cuppens, Ingénieur de Recherche du Centre d'Études et de Recherches de Toulouse (CERT) de l'Office National d'Études et des Recherches Aérospatiales (ONERA);
- Yves Deswarte, Directeur de Recherche INRIA au LAAS-CNRS;
- Laurent Dubief, Adjoint au Directeur Général, Inforsud;
- Claude Nodot, Docteur en informatique;
- Brian Randell, Professeur à l'Université de Newcastle-upon-Tyne;

pour l'honneur qu'ils me font en participant à mon jury. Je remercie tout particulièrement Jean-Jacques Quisquater et Pierre Rolin, qui ont accepté la charge d'être rapporteurs.

Ce travail n'aurait pas pu être effectué sans le soutien de l'UAP. Je tiens à remercier en particulier Jean-Jacques Duby pour avoir accepté de contribuer au financement de mes travaux, et Claude Nodot qui a eu la charge et, j'espère, le plaisir de suivre cette étude pendant plusieurs années.

Je voudrais également remercier le Crédit Agricole Mutuel, et plus particulièrement la Caisse Régionale Quercy-Rouergue, de m'avoir permis d'expérimenter les méthodes développées dans cette étude. Dominique Brunck et Raymond Rayssac

notamment ont joué un rôle important dans la réalisation de cette expérience. Enfin, je tiens à remercier l'ensemble du personnel de l'agence de Villefranche de Rouergue pour son accueil chaleureux.

J'exprime toute ma gratitude à Frédéric Cuppens pour m'avoir fait partager ses compétences, pour l'effort qu'il a fourni dans la relecture de ce travail, et enfin pour l'accueil amical qu'il m'a toujours réservé. Ses travaux m'ont permis de découvrir un domaine de la logique qui joue un rôle majeur dans ce mémoire.

J'exprime toute mon amitié à Marc Dacier pour m'avoir fait partager ses idées, ses résultats, et son enthousiasme dès le début de mon travail. Son travail fondateur sur l'évaluation quantitative de la sécurité est à l'origine du mien.

Je ne saurais oublier tous les membres du groupe TSF et du Laboratoire d'Ingénierie de la Sûreté de fonctionnement (LIS), permanents, doctorants et stagiaires, ainsi que Joëlle Penavayre et Marie-José Fontagne : leur sympathie et leur disponibilité contribuent à créer un cadre de travail très agréable. Je réserve ici une mention spéciale à Pascale Thévenod, Hélène Waeselynck et Yves Crouzet qui ont contribué à me faire découvrir la recherche, à Mathieu Béraud et Gaël Munduteguy qui m'ont apporté une aide significative lors d'un stage d'étude, et à Marie Borrel pour ses relectures constantes.

Mes remerciements s'adressent également à tous les membres des services *Informatique et Instrumentation*, *Documentation-Édition*, *Magasin*, *Entretien*, *Direction-Gestion*, et *Réception-Standard* qui m'ont toujours permis de travailler dans d'excellentes conditions. Le service 2I a même contribué directement à mes travaux.

Enfin, bien sûr, je remercie chaleureusement tous ceux qui, en dehors du laboratoire, m'ont accompagné et soutenu. Ils sont nombreux, à commencer par mes parents, et ce travail leur appartient à tous. La part qui revient à Marie est sans doute la plus substantielle.

TABLE DES MATIÈRES

INDEX DES FIGURES

INDEX DES TABLEAUX

Introduction générale

Ce mémoire est consacré à l'étude de la sécurité des systèmes d'information. À l'instar de [Dacier 1994], nous y soutenons l'idée selon laquelle il est nécessaire de se démarquer d'une vision binaire de la sécurité consistant à déclarer simplement qu'un système d'information est ou n'est pas sûr. Cette position se démarque de la vision traditionnelle de la sécurité, et elle conduit à adhérer aux méthodes permettant une évaluation quantitative de la sécurité, et s'intéressant avant tout à la sécurité opérationnelle des systèmes.

Les objectifs de notre étude sont :

- tout d'abord de développer l'approche d'évaluation quantitative de la sécurité des systèmes, et notamment des systèmes informatiques, en montrant son adéquation avec les besoins de sécurité réels existant dans ces systèmes ;
- et ensuite d'étendre cette approche vers les systèmes d'information au sens large, notamment vers les organisations conventionnelles.

Notre point de vue implique qu'une évaluation de la sécurité d'un système d'information ne peut pas être mise en œuvre sans une description explicite de ce qui est autorisé, ou légal, ou normatif ; car une telle description constitue le cadre de référence indispensable de l'évaluation. L'objet de notre thèse est la définition de ce cadre, et sa définition pour l'évaluation. En conséquence, le développement de cette approche demande :

- d'une part d'étudier les fondements des méthodes de définition des besoins de sécurité des systèmes d'information et la manière dont ces besoins trouvent leur incarnation dans ce que nous appellerons la politique de sécurité de ces systèmes ;
- et d'autre part d'établir le lien entre la définition de cette politique de sécurité et les méthodologies existantes d'évaluation quantitative de la sécurité permettant de tenir compte des vulnérabilités de ces systèmes qui vont à l'encontre des besoins de sécurité.

Nous proposons une méthode de spécification formelle pour définir la politique de sécurité d'un système d'information ou d'une organisation. Jusqu'à présent, dans de telles spécifications, la distinction entre le comportement normatif (tel qu'il *devrait être*) et le comportement réel (tel qu'il *est*) a été délaissée. Généralement, il est impossible de spécifier qu'un comportement du système est illégal mais néanmoins possible. Un comportement illégal est simplement éliminé par la spécification. Pourtant, il est utile de pouvoir décrire les conditions d'une telle évolution, et

les actions à mettre en œuvre quand un comportement illégal survient néanmoins. La logique modale, ou plus précisément la logique déontique, fournit un moyen d'obtenir une telle description par l'utilisation d'opérateurs spéciaux qui indiquent justement la nature d'un comportement: légal ou non. Un tel formalisme offre donc l'opportunité de décrire les besoins de sécurité d'une organisation non seulement du point de vue des propriétés de sécurité attendues, mais également du point de vue du fonctionnement effectif. Dans ce mémoire, nous présentons une méthode de spécification d'une politique de sécurité appuyée sur la logique déontique qui permet de décrire de façon naturelle et rigoureuse les besoins de sécurité des systèmes d'information.

La prise en compte du fonctionnement réel d'un système d'information suppose, du point de vue de la sécurité, d'accepter la présence de vulnérabilités résiduelles. Dans la position que nous soutenons, la constatation de l'existence de ces vulnérabilités ne conduit pas forcément à leur élimination. En effet, des contingences indépendantes des besoins de sécurité peuvent empêcher ou rendre indésirable cette élimination. Il s'agit alors d'évaluer l'impact de ces vulnérabilités sur la sécurité du système. Nous présentons donc une méthode d'évaluation quantitative de la sécurité, que nous développons, permettant, à partir de la politique de sécurité du système d'information, de fournir une indication objective de la confiance que l'on peut accorder à ce système du point de vue de la sécurité.

Afin de soutenir la méthodologie de spécification et d'évaluation de la sécurité exposée dans ce mémoire, nous présentons plusieurs exemples d'application.Deux exemples réels sont notamment développés tout au long de l'exposé pour illustrer les différents aspects de la méthode. Le premier correspond à la mise en œuvre de l'évaluation quantitative pendant une longue période sur un système informatique de grande taille possédant des mécanismes de sécurité simples mais dont on connaît un nombre significatif de vulnérabilités, un réseau de stations de travail sous UNIX. Le second est consacré à la définition de la politique de sécurité et à l'évaluation ponctuelle de deux objectifs de sécurité simples dans une organisation humaine de taille moyenne, une agence bancaire. Le premier exemple démontre notamment l'adéquation des résultats obtenus grâce à l'évaluation quantitative avec les besoins réels concernant la sécurité opérationnelle. Le second illustre la faisabilité de l'approche proposée dans le cadre d'une organisation.

La structure de ce mémoire est la suivante: le premier chapitre présente toute d'abord la terminologie que nous utilisons, et définit plus précisément le contenu d'une politique de sécurité. Ce chapitre est également consacré à l'étude des différentes approches existantes concernant la définition des modèles et des politiques de sécurité pour les systèmes d'information, et l'évaluation de la sécurité. Nous présentons ainsi les principaux modèles de sécurité utilisés pour décrire les mécanismes et les propriétés de sécurité des systèmes et notamment des systèmes informatiques. Nous étudions également les travaux actuels en matière d'évaluation de la sécurité, du point de vue des critères d'évaluation normalisés, de l'analyse des risques, et de l'évaluation quantitative.

Le deuxième chapitre est consacré à la présentation de la méthode de spécification formelle d'une politique de sécurité que nous proposons. Tout d'abord, le langage formel et la représentation semi-graphique utilisés sont définis et étudiés. Ensuite, plusieurs exemples sont présentés pour illustrer l'application de la méthode, d'abord pour la définition de la politique de sécurité d'une organisation réelle, puis pour la représentation de certains des modèles de sécurité présentés au premier chapitre.

Le troisième chapitre développe la méthode d'évaluation quantitative choisie. Tout d'abord, nous montrons comment on peut envisager de relier la spécification de la politique de sécurité obtenue précédemment au modèle sous-jacent aux mesures quantitatives de la sécurité: le graphe des privilèges. Ensuite, nous présentons différentes mesures qui peuvent être définies à partir du graphe des privilèges et nous étudions leurs propriétés. Enfin, nous poursuivons l'étude des exemples d'application pratique développés au second chapitre et nous analysons les résultats obtenus.

Enfin, une conclusion générale termine l'exposé de ces travaux, et présente les axes de recherche qui pourraient les prolonger.

Chapitre 1 Approches classiques de la sécurité

Dans ce chapitre, nous passons en revue les différentes approches de la sécurité représentées dans la littérature. Dans une première étape, nous présentons les concepts de base correspondant à l'étude de la sécurité en tant qu'attribut composite du domaine plus général de la sûreté de fonctionnement. Ensuite, nous examinons les principaux modèles de sécurité et les principales politiques de sécurité utilisés dans l'étude de la sécurité des systèmes informatiques. Enfin, nous présentons les différentes approches associées à l'évaluation, ou plus précisément à la validation, de la sécurité.

1.1 Concepts de base

Nous présentons dans cette section les concepts de base et la terminologie relatifs à la *sûreté de fonctionnement*, propriété générale des systèmes englobant les notions habituelles de fiabilité, disponibilité, sécurité, etc. La sécurité représente un attribut particulier de la sûreté de fonctionnement. Cette partie est adaptée de [Laprie 1995].

1.1.1 Concepts de sûreté de fonctionnement

1.1.1.1 Définitions

La **sûreté de fonctionnement** d'un système informatique est la propriété qui permet à ses utilisateurs de placer une *confiance justifiée* dans le service qu'il leur délivre. Le **service** délivré par un système est son comportement tel que perçu par son, ou ses utilisateurs; un **utilisateur** est un autre système (humain ou physique) qui interagit avec le système considéré.

Selon la, ou les applications auxquelles le système est destiné, l'accent peut être mis sur différentes facettes de la sûreté de fonctionnement, ce qui revient à dire que la sûreté de fonctionnement regroupe des propriétés différentes mais complémentaires, qui permettent de définir ses *attributs*:

- le fait d'être prêt à l'utilisation conduit à la **disponibilité**;
- la continuité de service conduit à la **fiabilité**;
- la non-occurrence de conséquences catastrophiques pour l'environnement conduit à la **sécurité-innocuité**;

- la non-occurrence de divulgations non-autorisées de l'information conduit à la **confidentialité**;
- la non-occurrence d'altérations inappropriées de l'information conduit à l'**intégrité**;
- l'aptitude aux réparations et aux évolutions conduit à la **maintenabilité**.

L'association de la confidentialité, de l'intégrité et de la disponibilité vis-à-vis des actions autorisées conduit à la **sécurité-confidentialité**.

Une **défaillance** du système survient lorsque le service délivré dévie de l'accomplissement de la fonction du système, c'est-à-dire ce à quoi le système est destiné. Une **erreur** est la partie de l'état du système qui est susceptible d'entraîner une défaillance: une erreur affectant le service est une indication qu'une défaillance survient ou est survenue. Une **faute** est la cause adjugée ou supposée d'une erreur.

Le développement d'un système sûr de fonctionnement passe par l'utilisation combinée d'un ensemble de moyens qui peuvent être classés de la manière suivante:

- **prévention des fautes**: comment empêcher l'occurrence ou l'introduction de fautes;
- **tolérance aux fautes**: comment fournir un service à même de remplir la fonction du système en dépit des fautes;
- **élimination de fautes**: comment réduire la présence (nombre, sévérité) des fautes;
- **prévision des fautes**: comment estimer la présence, la création et les conséquences des fautes.

Les notions qui ont été introduites peuvent donc être groupées en trois classes (figure 1):

- les **entraves** à la sûreté de fonctionnement: fautes, erreurs et défaillances; elles sont les circonstances indésirables — mais non inattendues — causes ou résultats de la non-sûreté de fonctionnement (dont la définition se déduit simplement de celle de la sûreté de fonctionnement: la confiance ne peut plus, ou ne pourra plus, être placée dans le service délivré);
- les **moyens** pour la sûreté de fonctionnement: prévention des fautes, tolérance aux fautes, élimination des fautes et prévision des fautes; il s'agit des méthodes et techniques permettant de fournir au système l'aptitude à délivrer un service conforme à l'accomplissement de sa fonction et de donner confiance dans cette aptitude;
- les **attributs** de la sûreté de fonctionnement: disponibilité, fiabilité, sécurité-innocuité, confidentialité, intégrité, maintenabilité; ils permettent d'exprimer les propriétés qui sont attendues du système, et d'apprécier la qualité du service délivré, telle que résultant des entraves et des moyens de s'y opposer.

La **spécification** du système, c'est-à-dire une description agréée de la fonction ou du service attendu du système, joue un rôle central pour la sûreté de fonctionnement. La fonction ou le service est habituellement décrit, ou spécifié, d'abord en termes de ce qui devrait être rempli ou délivré concernant la finalité première du

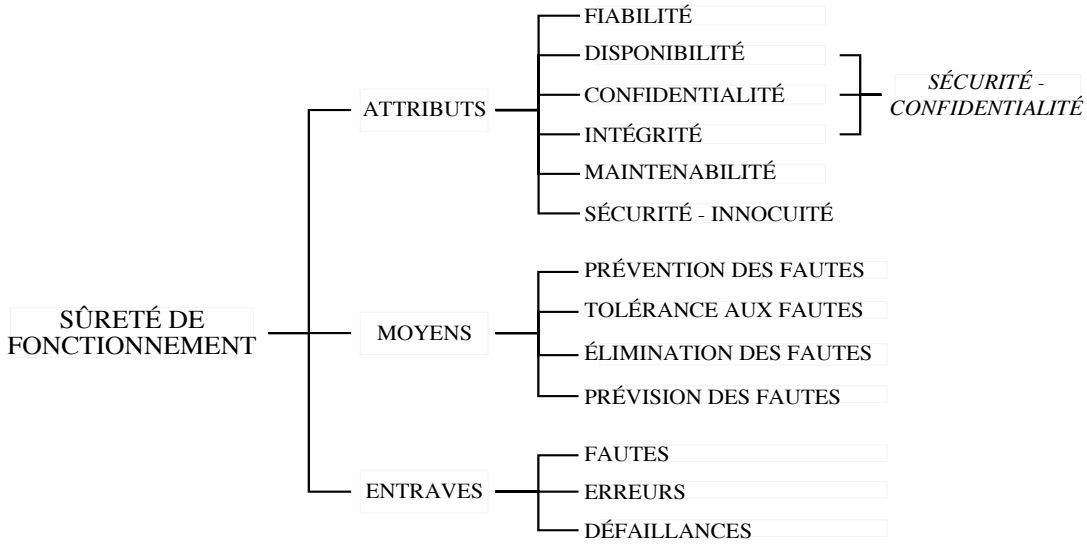


Figure 1 - L'arbre de la sûreté de fonctionnement

système (par exemple, effectuer des transactions, commander ou surveiller un procédé, piloter un missile,…). Lorsque l'on considère des systèmes de sécurité (soit sécurité-innocuité, soit sécurité-confidentialité), cette description est généralement complétée par l'énoncé de ce qui ne devrait pas arriver (par exemple, les états dangereux pouvant mener à une catastrophe, ou à la divulgation d'informations sensibles). Cette dernière description conduit à identifier des fonctions additionnelles que le système devrait remplir afin de réduire les possibilités d'occurrence des événements redoutés. Par exemple, en ce qui concerne la sécurité-confidentialité, une politique d'identification et une politique d'autorisation précisent les différentes méthodes qui doivent être utilisées pour authentifier un utilisateur et vérifier ses droits. Au travers de cette description, on identifie donc des fonctions nouvelles qui sont utilisées pour assurer des propriétés non-fonctionnelles de sécurité-confidentialité dans le système.

De plus, la spécification de ces fonctions peut:

- être exprimée selon divers points de vue et divers degrés de détail: spécification des besoins, spécification de conception, spécification de réalisation,…;
- être décomposée selon l'absence ou la présence de défaillances; le premier cas est relatif à ce qui est habituellement appelé le mode d'opération *nominal*, et le second peut être relatif à un mode d'opération *dégradé*, si les ressources survivantes ne sont plus suffisantes pour assurer le mode nominal.

En conséquence, il n'y a pas généralement une seule spécification, mais plusieurs, et un système peut défaillir par rapport à l'une de ces spécifications, tout en satisfaisant encore les autres. L'expression des fonctions d'un système est une activité qui est initiée naturellement dès les toutes premières étapes du développement de ce système, mais qui n'est généralement pas limitée à cette phase de la vie du système.

En effet, l'expérience montre que la spécification des fonctions du système se poursuit généralement tout au long de la vie du système, en raison de la difficulté à identifier ce qui est attendu d'un système.

La sécurité-confidentialité n'est pas introduite en tant qu'attribut de la sûreté de fonctionnement, conformément aux définitions habituelles associées à ce concept. Par exemple, la sécurité-confidentialité est définie dans [ITSEC 1991] comme une notion composite: "la combinaison de la *confidentialité*, prévention de la divulgation non autorisée de l'information, de l'*intégrité*, prévention de modification ou suppression non autorisée de l'information, et de la *disponibilité*, prévention de rétention non autorisée de l'information". L'ambiguïté résultant de l'utilisation du même terme, sécurité, pour désigner les deux notions différentes (mais non indépendantes) que sont la sécurité-innocuité et la sécurité-confidentialité correspond en fait à l'usage établi, d'où l'association d'un qualificatif pour lever cette ambiguïté: innocuité pour l'évitement de défaillances catastrophiques, confidentialité pour la prévention d'accès ou de manipulations non-autorisées de l'information. Ce dernier qualificatif est choisi en accord avec le constat que la confidentialité est la propriété la plus distinctive de la sécurité-confidentialité.

1.1.1.2 Les fautes dues à l'homme

Les fautes et leurs sources sont extrêmement diverses. Les cinq points de vue principaux selon lesquels on peut les classer sont leur cause phénoménologique (fautes physiques, fautes dues à l'homme), leur nature (fautes accidentelles, fautes intentionnelles), leur phase de création ou d'occurrence (fautes de développement, fautes opérationnelles), leur situation par rapport aux frontières du système (fautes internes, fautes externes), et leur persistance (fautes permanentes, fautes temporaires). Les définitions complètes associées à ce classement peuvent être trouvées dans [Laprie 1995].

Les fautes dues à l'homme donnent lieu à quatre classes de fautes combinées :

- les **fautes de conception**, qui sont des fautes de développement accidentelles ou intentionnelles sans volonté de nuire;
- les **fautes d'interaction**, qui sont des fautes externes, accidentelles ou intentionnelles sans volonté de nuire;
- les **logiques malignes**, qui sont des fautes internes intentionnellement nuisibles;
- les **intrusions**, qui sont des fautes opérationnelles externes intentionnellement nuisibles.

Quelques commentaires sur ces classes de fautes dues à l'homme sont particulièrement pertinents vis-à-vis de la sécurité-confidentialité:

1) Les fautes de conception intentionnelles sans volonté de nuire résultent généralement de compromis effectués durant la conception, dans un souci de conserver au système un niveau de performances acceptable ou de faciliter son utilisation, ou encore pour des raisons économiques; cependant ces fautes peu-

vent être des sources d'atteinte à la sécurité-confidentialité, sous la forme de *canaux cachés*. Un **canal caché** désigne un mécanisme non prévu pour la communication qui peut être utilisé pour transférer de l'information d'une manière qui viole la politique de sécurité du système d'information [ITSEC 1991].

2) Les fautes d'interaction intentionnelles sans volonté de nuire peuvent résulter de l'action d'un opérateur soit destinée à faire face à une situation imprévue, soit violant délibérément des procédures sans avoir réalisé les conséquences malheureuses de cette action. Généralement, ces fautes intentionnelles sans volonté de nuire ne sont identifiées comme des fautes qu'après qu'elles aient causé un comportement inacceptable du système, donc une défaillance.
3) Les logiques malignes recouvrent aussi bien des fautes de développement comme les chevaux de Troie, les portes dérobées, les bombes logiques ou temporelles, ou des fautes opérationnelles (pour le système considéré) comme les virus et les vers. Ces fautes peuvent être définies comme suit :
   - une **bombe logique** est une partie de programme qui reste dormante dans le système hôte jusqu'à ce qu'une date ou un événement d'activation survienne, ou que certaines conditions soient remplies, entraînant des effets dévastateurs pour le système hôte;
   - un **cheval de Troie** est un programme effectuant une fonction illicite tout en donnant l'apparence d'effectuer une fonction légitime; la fonction illicite peut être de divulguer ou de modifier des informations (attaque contre la confidentialité ou l'intégrité), ou peut être une bombe logique;
   - une **porte dérobée** est un moyen de contourner les mécanismes de contrôle d'accès; il s'agit d'une faille du système de sécurité due à une faute de conception accidentelle ou intentionnelle (cheval de Troie en particulier);
   - un **virus** est un segment de programme qui, lorsqu'il s'exécute, se reproduit en s'adjoignant à un autre programme (du système ou d'application), qui devient ainsi un cheval de Troie ; un virus peut être porteur d'une bombe logique;
   - un **ver** est un programme autonome qui se reproduit et se propage à l'insu des utilisateurs; un ver peut également être porteur d'une bombe logique.
4) Les intrusions ne peuvent être couronnées de succès sans l'existence de fautes de conception; on peut identifier des similarités évidentes et néanmoins intéressantes entre une intrusion et une faute accidentelle externe "exploitant" un défaut de blindage; il faut de plus noter que le caractère externe des intrusions n'exclut pas qu'elles soient tentées par des opérateurs ou administrateurs du système qui abusent de leur pouvoir.

Les définitions complètes des différentes notions relatives à la sûreté de fonctionnement, incluant certains aspects que nous n'avons pas évoqués ici (notamment en ce qui concerne les moyens de la sûreté de fonctionnement) pourront être trouvées dans [Laprie 1995].

1.1.2 La sécurité-confidentialité

Lorsque le contexte permet de lever aisément le doute, l'adjonction d'un qualificatif permettant de distinguer entre la sécurité-confidentialité et la sécurité-innocuité n'est pas nécessaire. Dans la suite de ce mémoire, nous utiliserons donc directement le terme "sécurité" pour désigner la sécurité-confidentialité, qui fait l'objet de nos travaux.

Assurer la sécurité d'un système consiste à garantir le maintien d'un certain nombre de propriétés de sécurité définissant les caractéristiques de confidentialité, d'intégrité et de disponibilité qui doivent être maintenues dans le système. Ceci implique d'empêcher la réalisation d'opérations illégitimes contribuant à mettre en défaut ces propriétés, mais aussi de garantir la possibilité de réaliser des opérations légitimes dans le système. La description de ces différentes propriétés fait partie de la **politique de sécurité** du système. Assurer la sécurité du système, c'est donc assurer que les propriétés retenues sont vérifiées (et donc garantir la non-occurrence de défaillances vis-à-vis de ces propriétés).

Il faut noter qu'a priori, l'occurrence d'opérations illégitimes n'est pas forcément le signe d'une action intentionnellement nuisible d'un utilisateur. Des opérations mettant en danger la sécurité du système peuvent survenir du fait d'un utilisateur autorisé et bien intentionné, mais qui ignore certaines des propriétés attendues du système ou qui ne maîtrise pas complètement toutes les implications des opérations qu'il effectue.

1.1.2.1 Confidentialité

La **confidentialité** peut être définie comme la prévention de la divulgation non-autorisée de l'information [ITSEC 1991, §6.18]. Ceci correspond à empêcher un utilisateur de consulter directement une information qu'il n'est pas autorisé à connaître, mais aussi à empêcher un utilisateur autorisé à lire une information de la divulguer à un utilisateur non autorisé à y accéder. Le terme information doit être pris au sens le plus large : il recouvre non seulement les données elle-mêmes, mais aussi les flux d'information et la connaissance de l'existence des données ou des communications. Assurer la confidentialité d'un système est donc une tâche bien plus compliquée qu'il n'y paraît de prime abord. Il faut analyser tous les chemins qu'une information particulière peut prendre dans le système pour s'assurer qu'ils sont sécurisés. Il importe également de prendre en compte les connaissances qu'un ou plusieurs utilisateurs peuvent déduire à partir des informations qu'ils acquièrent (soit parce que le système les autorise à les consulter, soit parce qu'elles font partie des informations librement diffusées dans l'environnement du système). Il faut donc contrôler non seulement les informations présentes dans le système, mais aussi les liens logiques qui peuvent les relier entre elles ou à des informations publiques, afin de garantir que les informations diffusées par le système ne transforment pas une information protégée en secret de polichinelle [Morgenstern 1988 ; Garvey & Lunt 1992 ; Cuppens & Trouessin 1994].

Les attaques contre la confidentialité consistent à essayer d'obtenir des informations qui doivent être protégées selon la politique de sécurité en dépit des moyens de protection et des règles de sécurité. Par exemple, les écoutes passives consistent à accéder aux données transmises sur un canal de communication (câble réseau) ou stockées sur un support vulnérable (disques externes, bandes magnétiques). Une telle écoute peut, dans certaines circonstances, permettre d'accéder à des informations sensibles, comme par exemple le mot de passe d'un utilisateur, tapé sur un terminal connecté par une liaison série à un ordinateur central, et qui transite en clair entre ce terminal et la machine. On voit également que cette attaque peut être particulièrement difficile à identifier a posteriori étant donné l'absence totale de traces laissées dans le système.

1.1.2.2 Intégrité

L'intégrité peut être définie comme la capacité du système à empêcher la corruption des informations par les fautes accidentelles ou intentionnelles, et à garantir leur mise à jour correcte. Dans le cas des fautes intentionnelles, les attaques contre l'intégrité visent soit à introduire des fausses informations, soit à modifier ou à détruire des informations pour que le service inapproprié délivré par le système produise un bénéfice pour l'attaquant. C'est par exemple le cas pour une fraude. Il faut noter qu'en général, les erreurs introduites par l'attaquant ne sont pas facilement détectables, c'est-à-dire que l'information *paraît* intègre alors qu'elle ne l'est pas, ce qui rend les défaillances résultantes difficilement détectables. Dans le cas où une information est visiblement erronée, les méthodes permettant de parer à la corruption des informations due à des fautes accidentelles s'appliquent plus naturellement et la défaillance résultante est généralement mieux maîtrisée.

Afin de se prémunir contre les fautes affectant l'intégrité des données, il importe d'intégrer dans le système des mécanismes permettant de rendre détectables des erreurs affectant ces données. Dans le cas de fautes accidentelles, comme par exemple celles pouvant affecter le support physique d'une voie de communication d'informations, l'adjonction de codes correcteurs aux données transmises peut être suffisante pour détecter une donnée erronée. Néanmoins, quand il s'agit de se prémunir contre les altérations malveillantes de l'information, ceci est généralement insuffisant puisque l'attaquant est en mesure d'altérer également le code accompagnant l'information pour lui donner une apparence d'intégrité. Dans ce cas, il faut donc séparer les voies de transmission de la signature de l'information et de l'information elle-même, ou utiliser des mécanismes plus sophistiqués comme les algorithmes cryptographiques permettant la génération d'un sceau ou d'une signature.

1.1.2.3 Disponibilité

Une attaque contre un système peut avoir simplement pour but d'empêcher le système de remplir le service pour lequel il a été conçu. Il s'agit alors d'une attaque contre la disponibilité du système ou **déni de service**. Ces attaques consistent à faire en sorte que les actions du système ne correspondent plus à ce que l'on attend

de lui, soit parce que le résultat des actions effectuées par le système est erroné, soit parce que ce résultat n'est pas disponible en temps voulu. La première catégorie d'attaque est étroitement liée à l'intégrité étant donné qu'elle consiste à modifier l'information présente dans le système cible, ou dans son système de communication, voire au cours de son traitement, afin que le système fournisse un résultat incorrect. La deuxième catégorie peut également trouver sa source dans une attaque contre l'intégrité des données ou du système dont l'objectif est d'interrompre le traitement de l'information (ou tout au moins de le retarder), comme dans le cas de la destruction d'un lien de communication. Cependant ce type d'attaque peut également être mis en œuvre en perturbant le fonctionnement temporel du système, par exemple en surchargeant certaines des ressources dont il est dépendant, ou en surchargeant le système lui-même, afin que le temps de traitement d'une opération particulière devienne inacceptable.

Il faut noter que, en ce qui concerne les systèmes informatiques, les attaques contre la disponibilité d'un système semblent constituer à l'heure actuelle un des problèmes de sécurité majeurs. En effet, les systèmes ne sont pas conçus pour résister à de telles attaques. De plus, celles-ci sont facilitées par de nombreuses vulnérabilités introduites dans le système pour en améliorer les performances dans des circonstances nominales. La facilité de mise en œuvre de ce type d'attaque, notamment vis-à-vis des systèmes informatiques, est un problème préoccupant.

1.2 Politiques et modèles de sécurité

La politique de sécurité d'un système constitue une partie de la spécification des propriétés non-fonctionnelles de ce système. Plus précisément, elle précise les différentes propriétés qui sont relatives aux différents attributs de sécurité-confidentialité: l'intégrité, la disponibilité et la confidentialité.

1.2.1 Contenu d'une politique de sécurité

Dans les [ITSEC 1991, §2.9], la **politique de sécurité** *"est l'ensemble des lois, règles et pratiques qui régissent la façon dont l'information sensible et les autres ressources sont gérées, protégées et distribuées à l'intérieur d'un système spécifique"*.

Plus précisément, on peut considérer qu'une politique de sécurité définit:

- des **objectifs de sécurité**, c'est-à-dire des propriétés de confidentialité, d'intégrité et de disponibilité désirées pour le système;
- et des **règles de sécurité** permettant de modifier l'état de sécurité du système, qui sont imposées au comportement du système dans le but d'obtenir ces propriétés.

La politique de sécurité est **cohérente** si, partant d'un état sûr où les propriétés sont satisfaites, il n'est pas possible, sans violer les règles, d'atteindre un état non-sûr où ces propriétés ne sont pas satisfaites. La définition des propriétés de sécurité peut

nécessiter la description d'éléments appartenant en propre au système (des sujets ou des objets spécifiques, les opérations mises en œuvre, ou une représentation de l'organigramme d'une organisation, par exemple). Ces **éléments de description**, présents dans la politique de sécurité, constituent une description partielle du système. Contrairement à eux, les objectifs et les règles de sécurité sont relatifs aux besoins de sécurité du système. Les objectifs de sécurité décrivent les propriétés attendues et définissent ainsi la notion d'état sûr pour le système. La formulation de ces objectifs peut utiliser des notions comme la permission, l'interdiction, ou l'obligation [Minsky & Lockman 1985], et décrit la manière dont elles s'appliquent au système. Les règles de sécurité s'attachent plus particulièrement à décrire (à un haut niveau) les mécanismes fondamentaux de sécurité utilisés dans le système. Si des attributs spécifiques à la sécurité sont introduits explicitement, ces règles spécifient la manière dont ils sont manipulés. L'ensemble des règles de sécurité constitue donc une description des moyens permettant de modifier l'état de sécurité du système. Certaines règles peuvent également être introduites dans le but de contribuer à assurer la validité des propriétés de sécurité attendues du système.

La notion de politique de sécurité peut se développer dans trois directions distinctes: les politiques de sécurité **physique**, **administrative** et **logique**.

La première précise tout ce qui touche à la situation physique du système à protéger. En particulier, y sont définis les éléments critiques et les mesures prises vis-à-vis du vol par effraction, des agressions sur les personnes, des catastrophes naturelles, du feu, etc. De par la nature des éléments auxquels elle s'applique, une politique de sécurité physique s'attache avant tout à la description de certains objets physiques présents dans le système et au choix des objectifs. Dans le cas où les objectifs de sécurité ne sont pas satisfaits, une intervention directe sur le système (par exemple le renforcement d'un blindage, ou l'installation d'une temporisation sur un coffre-fort) est préalablement nécessaire. Sinon, la propriété recherchée n'est tout simplement pas assurée.

La politique de sécurité administrative est un ensemble de procédures qui traitent de tout ce qui ressort de la sécurité d'un point de vue organisationnel au sein de l'entreprise. La structure de l'organigramme choisi, la répartition des tâches et des responsabilités en différentes fonctions, et finalement l'affectation de ces fonctions aux individus en font également partie. Les propriétés de sécurité recherchées visent par exemple à limiter les cumuls ou les délégations abusives de pouvoirs, ou à garantir une séparation des pouvoirs.

La politique de sécurité logique s'intéresse quant à elle au contenu du système informatique ou, plus généralement, du système d'information présent dans l'organisation. Elle décrit les contrôles d'accès logiques, en spécifiant à qui il est permis d'accéder à quoi et dans quelles circonstances. Une politique de sécurité logique peut être raffinée en plusieurs branches correspondant aux différentes étapes de l'accès au système d'information. Un individu qui utilise le système soumis à la politique de sécurité doit d'abord s'identifier, et ensuite doit apporter la preuve qu'il est bien la personne qu'il prétend être. À ces deux phases correspondent la **politi-**

que d'identification et la **politique d'authentification**. Une fois ces étapes franchies, la **politique d'autorisation** définit les opérations que l'utilisateur est autorisé à réaliser dans le système. Dans le cadre d'une politique d'autorisation, les règles de sécurité constituent le **schéma d'autorisation** du système.

1.2.2 Politiques d'autorisation

1.2.2.1 Politiques de sécurité

Les politiques de sécurité s'attachent donc à définir des propriétés de sécurité désirées pour le système. Dans le cas des organisations, ces politiques de sécurité s'expriment généralement en langage naturel et constituent un **règlement de sécurité**, spécifiant simultanément les propriétés désirées et les règles devant être appliquées par les individus afin de garantir ces propriétés. On trouve ce type de règlement dans la plupart des organisations qui doivent respecter des contraintes de sécurité, que ce soit par obligation légale ou pour des raisons déontologiques (par exemple dans le domaine bancaire, dans le domaine des assurances, dans le domaine de l'informatique vis-à-vis des bases de données nominatives [CNIL 1978], ou dans le domaine médical [CO 1979; Saury 1991; CNIL 1994; DGS 1995]). De par leur nature informelle, ces règlements sont souvent sujets à interprétation. Ils peuvent contenir des ambiguïtés et conduire à des contradictions (vis-à-vis du secret médical par exemple, ces contradictions sont actuellement préoccupantes [Saury 1991; Abecassis 1995]; en ce qui concerne le respect de l'assentiment du patient d'autres difficultés ont également été identifiées [Pigeaud 1993]). Néanmoins, ils constituent vraisemblablement la grande majorité des politiques de sécurité en application, et ont fait l'objet de représentations formelles [Kanger 1972; Jones & Sergot 1992; Jones & Sergot 1993; Royakkers 1994; Cuppens & Saurel 1996; Cholvy & Cuppens 1997].

Les politiques d'autorisation utilisées dans les systèmes informatiques constituent la majorité des travaux possédant une base rigoureuse. Afin d'exprimer les objectifs de sécurité, ces politiques d'autorisation introduisent des éléments de modélisation particuliers. En effet, elles impliquent généralement une division de très haut niveau du système entre ses entités actives (appelées les sujets) et passives (appelées les objets), l'introduction d'attributs de sécurité associés aux éléments du système (comme dans le cas des politiques multi-niveaux), ou éventuellement l'utilisation d'une méthode spécifique de modélisation du système (comme dans le cas des politiques de contrôle de flux).

1.2.2.2 Modèles de sécurité

Cependant, les différentes classes de modèles de sécurité présentées dans la littérature correspondent souvent étroitement à la définition d'une politique de sécurité particulière: par exemple, les modèles basés sur la notion de treillis sont en liaison étroite avec la définition d'une politique multi-niveaux.

L'utilisation d'un **modèle de sécurité** pour la représentation de la sécurité a pour objectif de définir clairement en langage mathématique ce que décrit la politique de sécurité. Ainsi, les différentes politiques de sécurité étudiées dans la littérature ont également donné lieu à la définition de différentes classes de modèles de sécurité. Par exemple, une politique multi-niveaux est en liaison étroite avec les modèles basés sur la notion de treillis. Toutefois, la définition d'un modèle de sécurité n'est pas seulement guidée par le besoin d'une représentation formelle des objectifs de sécurité et des règles du schéma d'autorisation. Le choix des différents éléments du système pris en compte, des attributs spécifiques de la sécurité qui sont introduits, ainsi que des principes de construction qui apparaissent dans le modèle de sécurité, est également motivé par la nécessité de concevoir un modèle commode. Les critères qui conduisent à adopter un modèle particulier viennent principalement de deux sources. D'une part, les problèmes d'application d'un modèle de sécurité à un système particulier, par exemple un système informatique, motivent l'utilisation de représentations qui tiennent compte des contraintes de mise en œuvre. D'autre part, les propriétés de sécurité recherchées (les objectifs de sécurité) doivent être vérifiables ou tout au moins exprimables grâce au modèle utilisé. En définitive, le modèle de sécurité fournit un formalisme permettant de représenter de façon claire et non-ambiguë ce que signifie la sécurité pour le système considéré. Ce formalisme peut également servir de support à la vérification de la cohérence de la politique de sécurité pour ce système. Toutefois, il est possible que la politique de sécurité définie ne puisse pas être vérifiée pour un système particulier à l'aide du modèle considéré. La définition du comportement du système quand les propriétés attendues ne sont pas obtenues est aussi une préoccupation qui peut être prise en compte dans le modèle de sécurité. Toutefois, cette situation est ignorée par la plupart des modèles classiques présentés ci-après.

1.2.3 Politiques d'autorisation obligatoire et discrétionnaire

Les politiques d'autorisation peuvent se classer en deux grandes catégories : les **politiques discrétionnaires** et les **politiques obligatoires**. Dans les deux cas, on effectue en général une partition des éléments présents dans le système en deux grandes catégories : les entités actives ou **sujets** (individus, processus, etc.) qui manipulent l'information, et les entités passives ou **objets** (documents, fichiers, etc.) qui contiennent l'information.

Dans le cas d'une politique discrétionnaire, chaque objet est associé à un sujet précis responsable de l'information contenue dans cet objet (généralement, le propriétaire) qui peut manipuler librement les droits d'accès de cet objet, à sa *discrétion*. Le propriétaire de l'information peut donc librement définir et transmettre ces droits à lui-même ou un autre utilisateur. La gestion des accès aux fichiers du système d'exploitation UNIX constitue un exemple classique de mécanismes de contrôles d'accès basés sur une politique discrétionnaire. Sur ce système, trois types d'accès sont définis : *read* (consultation/lecture), *write* (modification/écriture) et *execute* (exécution), et ces droits peuvent s'appliquer soit au propriétaire du fichier, soit à

un groupe d'utilisateurs, soit aux autres utilisateurs. Mais supposons qu'un utilisateur u_1, propriétaire d'un fichier f_1, fasse confiance à un utilisateur u_2 mais pas à un utilisateur u_3. u_1 donne alors un droit d'accès en lecture à u_2 sur le fichier f_1, mais pas à u_3. Toutefois, u_2 est alors en mesure de faire une copie des informations contenues dans f_1 dans un autre fichier f_2 dont il est lui-même propriétaire. Dans ce cas, u_2 est alors aussi en mesure de donner à u_3 un droit en lecture sur cette copie. Ceci constitue une fuite d'informations qu'il est impossible, avec une politique d'autorisation discrétionnaire, de contrôler. De même, une politique discrétionnaire ne permet pas de résoudre le problème des chevaux de Troie. Un cheval de Troie est un programme qui, sous couvert de réaliser une action légitime, réalise à l'insu de la personne qui l'utilise une autre action qui peut consister en une attaque contre les règles de sécurité du système. Par exemple, sous UNIX, si l'on réussit à remplacer le programme standard d'authentification d'un nouvel utilisateur `login` par un programme spécifique, un utilisateur qui se connecte, pensant exécuter la véritable procédure d'authentification, confie son mot de passe à un programme qui peut par exemple le communiquer à un autre utilisateur.

Pour résoudre ce type de problème, les politiques obligatoires imposent, en plus des règles discrétionnaires, des règles incontournables destinées à assurer le respect de propriétés globales. Par exemple, il peut être précisé que des attributs de sécurité doivent être associés à chaque unité d'information dans le système, que ces attributs doivent être propagés à chaque manipulation ou création d'information, et que seuls les utilisateurs explicitement associés à un niveau de sécurité sont en mesure de manipuler ou d'accéder à une information de ce niveau. Ces règles obligatoires permettent de garantir que le système maintient une propriété globale de sécurité (confidentialité ou intégrité par exemple). Elles viennent s'ajouter aux règles du contrôle discrétionnaire (qui fournissent un principe commode pour régir les droits d'accès) et un utilisateur sera alors autorisé à manipuler une information si les droits correspondants lui ont été attribués (contrôle discrétionnaire) et s'il est habilité à le faire (contrôle obligatoire).

Des exemples classiques de politiques obligatoires sont la politique du DoD (Department of Defense) formalisée par Bell et LaPadula [Bell & LaPadula 1975], qui vise à offrir des propriétés de confidentialité, la politique de Biba [Biba 1977] basée sur les mêmes principes mais visant à offrir des propriétés d'intégrité, ou celle de Clark et Wilson [Clark & Wilson 1987] qui formalise des principes plus spécifiques à des systèmes commerciaux. Ces différents exemples sont détaillés dans les sections suivantes.

1.2.4 Modélisation des politiques discrétionnaires

Un certain nombre de modèles ont été définis afin de représenter dans un formalisme mathématique les mécanismes associés aux politiques de sécurité discrétionnaires. Nous présentons dans cette section les principaux modèles rencontrés dans la littérature. On notera que ces modèles sont des modèles généraux, qui peuvent

également représenter les propriétés rencontrées dans une politique de sécurité obligatoire. Toutefois, les politiques obligatoires sont généralement décrites par des modèles spécifiques.

1.2.4.1 Modèles basés sur les matrices de contrôle d'accès

La notion de matrice de contrôle d'accès, dédiée à la représentation des droits d'accès (autorisations), a d'abord été introduite par [Lampson 1971]. La structure de ce modèle est celle d'une machine à états où chaque état est un triplet (S,O,M), où S est un ensemble de sujets, O un ensemble d'objets (S étant un sous-ensemble de O), et M est une matrice de contrôle d'accès. La matrice M possède une ligne pour chaque sujet, une colonne pour chaque objet, et est telle que $M(s,o)$ est l'ensemble des droits d'accès que le sujet s possède sur l'objet o. Ces droits d'accès sont pris dans un ensemble fini A, défini par la politique de sécurité, et correspondent aux différentes opérations qu'un sujet peut réaliser sur un objet. La matrice des droits d'accès n'est pas figée. Elle évolue dans le temps, en fonction de la création de nouveaux sujets, de nouveaux objets et en fonction des opérations effectuées par les utilisateurs. L'état de sécurité est ainsi changé par toutes les actions qui modifient la matrice M.

En règle générale, la plupart des approches basées sur cette notion ajoutent des lignes et des colonnes à la matrice de contrôle d'accès à chaque fois qu'un nouveau processus agissant pour le compte d'un utilisateur est introduit dans le système ou qu'un nouveau fichier est créé. Ces lignes ou ces colonnes sont initialisées avec un certain nombre de valeurs par défaut fixées par les fichiers de configuration de l'utilisateur. Un utilisateur peut ultérieurement modifier les droits d'accès des fichiers qu'il a créé (dans une politique de sécurité discrétionnaire), mais n'effectue pas directement des modifications dans la matrice M. En effet, les opérations de modification des droits d'accès doivent être légitimes, et peuvent aussi être soumises à des contrôles imposés par le schéma d'autorisation (dans une politique de sécurité obligatoire), et l'utilisateur réalise ces opérations par le biais d'utilitaires du système qui n'effectuent les modifications demandées que si elles sont conformes à ce schéma d'autorisation.

Ce modèle basé sur les matrices de contrôle d'accès a connu une longue évolution et a été progressivement amélioré, notamment en raison des travaux initiaux de Bell et LaPadula [Bell & LaPadula 1975], et de Harrison, Ruzzo et Ullman [Harrison *et al*. 1976].

1.2.4.1.1 Le modèle HRU

Harrison, Ruzzo et Ullman ont utilisé le modèle de la matrice de contrôle d'accès de Lampson dans le but d'étudier la complexité de la tâche de vérification des propriétés assurées par une politique d'autorisation. Pour préciser le contexte de leurs

travaux, ils ont considéré un modèle de sécurité particulier, le modèle *HRU*, analogue au modèle de Lampson, mais contenant seulement des commandes de modification de la matrice *M* de la forme suivante:

command $\alpha(x_1, x_2, \ldots, x_k)$
  if $a' \in M(s', o')$ and $a'' \in M(s'', o'')$ and … and $a^{(m)} \in M(s^{(m)}, o^{(m)})$
  then $op_1; op_2; \ldots; op_n$
end

Tableau 1 - Format d'une commande HRU

où $a^{(i)} \in A$, x_i est un paramètre de la commande, et chaque op_i est une des opérations élémentaires suivantes (dont la sémantique est naturellement conforme à la dénomination):

enter *a* **into** *M*(*s*,*o*)	**delete** *a* **from** *M*(*s*,*o*)
create subject *s*	**destroy subject** *s*
create object *o*	**destroy object** *o*

Tableau 2 - Opérations élémentaires de HRU

Étant donné un système, une configuration initiale Q_0, et un droit *a*, on dit que Q_0 est *sûr* pour *a* s'il n'existe aucune séquence d'opérations qui, exécutée à partir de l'état Q_0, peut amener le droit *a* dans une cellule de la matrice de contrôle d'accès dans laquelle *a* ne se trouve pas déjà. La démonstration de cette propriété constitue le **problème de protection** (*"safety problem"*). Harrison, Ruzzo et Ullman ont prouvé deux théorèmes fondamentaux concernant la complexité du problème de protection [Harrison *et al*. 1976]:

- le problème de protection est *indécidable* dans le cas général;
- le problème de protection est *décidable* pour les **systèmes à mono-opération**, qui sont les systèmes dans lesquels toutes les commandes ne contiennent qu'une seule opération élémentaire.

En imposant d'autres contraintes sur les différentes commandes utilisables dans le système, plusieurs autres démonstrations de décidabilité ont également pu être obtenues [Harrison & Ruzzo 1978; Lipton & Snyder 1978]. Néanmoins, dès leur présentation dans le cadre du modèle *HRU*, ces deux premiers théorèmes ont clairement identifié la problématique de la sécurité. D'une part, le modèle *HRU* sans contraintes peut exprimer une grande variété de politiques de sécurité, mais il n'y a en général aucun moyen de vérifier les propriétés de ces politiques. D'autre part, même s'il est plus commode à manipuler, le modèle *HRU* à mono-opération est trop simple pour exprimer des politiques de sécurité intéressantes dans la pratique. Par exemple, un système à mono-opération ne permet pas d'exprimer des politiques d'autorisation dans lesquelles les sujets qui créent des objets se voient attribuer des droits spécifiques sur ces objets puisqu'il n'y a pas d'opération élémentaire qui permette simultanément de créer un objet et d'y associer des droits.

1.2.4.1.2 Le modèle Take-Grant

Divers développements issus du modèle *HRU* ont été réalisés par la suite dans le but de déterminer un modèle de sécurité suffisamment expressif pour représenter des politiques d'autorisation sophistiquées, mais qui reste néanmoins facile à manipuler mathématiquement.

Le modèle *Take-Grant*, introduit par [Jones *et al*. 1976], est une première variante du modèle *HRU*, obtenue par restriction des différentes commandes utilisables. Celles-ci se répartissent en quatre grandes catégories:

- les commandes de type *create* qui permettent de créer un objet et d'attribuer initialement un droit d'accès à un sujet sur cet objet;
- les commandes de type *remove* qui permettent de retirer un droit d'accès d'un sujet sur un objet;
- les commandes de type *grant* qui permettent à un sujet possédant déjà un droit d'accès sur un objet ainsi que le droit spécial *g* sur un autre sujet de céder ce droit d'accès à ce dernier sujet;
- les commandes de type *take* qui permettent à un sujet possédant le droit spécial *t* sur un autre sujet d'obtenir les droits d'accès que ce sujet possède sur les objets.

Ce modèle de sécurité introduit également une représentation graphique pour représenter le contrôle d'accès. Les nœuds du graphe sont de deux types: ils représentent les sujets ou les objets. Les arcs du graphe sont étiquetés, et ces étiquettes appartiennent à l'ensemble *A* des droits. La figure 2 présente un exemple de graphe représentant un état de protection simple contenant deux sujets *P* et *R*, un objet *O*, et un droit d'accès élémentaire α. Les modifications de l'état de sécurité du système sont représentées par des réécritures du graphe. La représentation graphique des quatre règles de réécriture correspondant aux quatre catégories de commandes présentées précédemment est donnée dans la figure 3.

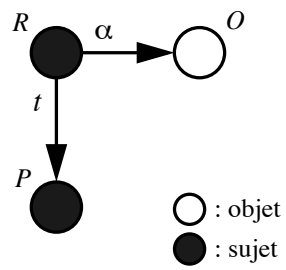


Figure 2 - Un exemple simple d'état de protection

Ces quatre grandes catégories conduisent à définir quatre nouvelles commandes pour chaque droit d'accès élémentaire pris en compte dans la politique d'autorisation. Les droits d'accès spéciaux *t* et *g*, ainsi que les règles de type *take* et *grant* associées, correspondent à des règles supplémentaires imposées au niveau du schéma d'autorisation et qui régissent l'évolution de l'état de sécurité du système (c'est-à-dire de la matrice de contrôle d'accès). L'existence de ces nouvelles règles permet de garantir que le modèle *Take-Grant* possède un algorithme de décision de complexité linéaire pour le problème de protection. Toutefois, les hypothèses sous-jacentes à ce modèle sont assez peu réalistes, en effet, les propriétés démontrables correspondent à une hypothèse de pire cas sur le comportement des utilisateurs

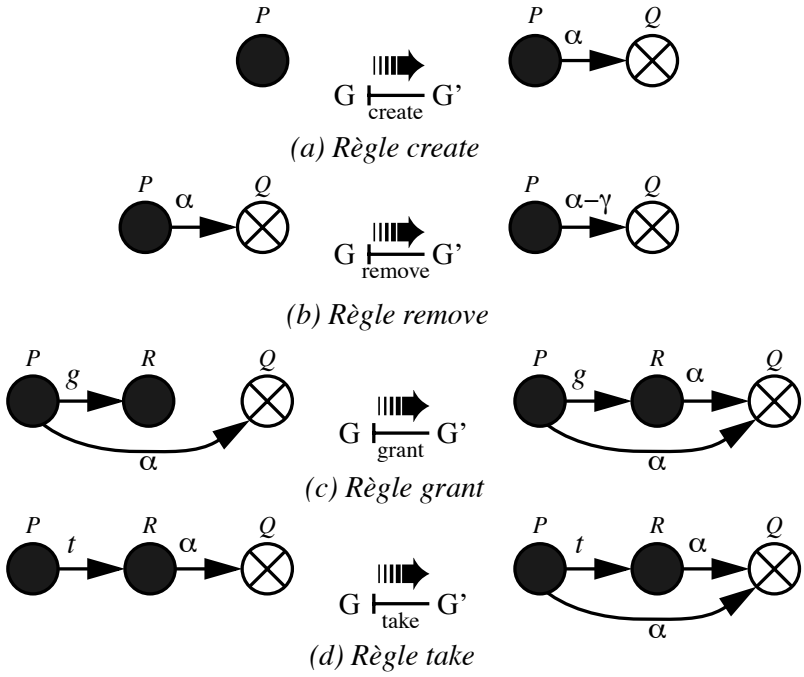


Figure 3 - Les règles de réécriture du modèle *Take-Grant*[a]

a. Dans ces figures, les nœuds marqués d'une croix peuvent représenter soit des sujets, soit des objets.

du système vis-à-vis des objectifs de sécurité (c'est-à-dire le cas où tous les utilisateurs sont susceptibles de collaborer à la mise en défaut des objectifs). Plusieurs raffinements des propriétés démontrables grâce au modèle *Take-Grant* ont été proposés, notamment afin de lever cette hypothèse et de se concentrer sur les cas où un utilisateur [Snyder 1981] ou un ensemble de plusieurs utilisateurs [Dacier 1993; Dacier 1994] tentent de mettre en défaut les objectifs de sécurité. L'étude des propriétés de ce modèle dans des cadres spécifiques à également fait l'objet de nombreux travaux, recensés dans [Dacier 1994].

1.2.4.1.3 TAM

Plus proche du modèle *HRU*, le modèle *SPM* (*Schematic Protection Model*) de [Sandhu 1988; Sandhu 1992], qui contient des types pour la sécurité, possède un sous-ensemble décidable plus étendu que le modèle *Take-Grant*. Ce modèle est également à l'origine du modèle *TAM* (*Typed Access Matrix*). *TAM* est obtenu en introduisant un typage fort dans le modèle *HRU*. Si T est l'ensemble de tous les types définis dans le système, T_S un ensemble de types pour les sujets du système

(avec $T_S \subset T$) et T_O un ensemble fini de types pour les objets du système (avec $T_O = T - T_S$) les opérations primitives de *TAM* sont alors (avec $t_s \in T_S$ et $t_o \in T_O$):

enter a **into** $M(s,o)$	**delete** a **from** $M(s,o)$
create subject s **of type** t_s	**destroy subject** s **of type** t_s
create object o **of type** t_o	**destroy object** o **of type** t_o

Tableau 3 - Opérations élémentaires de *TAM*

On peut définir les différentes commandes utilisables par ce modèle d'une manière analogue au modèle *HRU* ($x_i{:}t_i$ signifiant que le paramètre x_i est du type t_i):

command $\alpha(x_1{:}t_1, x_2{:}t_2, \ldots, x_k{:}t_k)$
  if $a' \in M(s', o')$ and $a'' \in M(s'', o'')$ and … and $a^{(m)} \in M(s^{(m)}, o^{(m)})$
  then $op_1;op_2;\ldots;op_n$
end

Tableau 4 - Format d'une commande *TAM*

Tout comme le modèle *HRU*, *TAM* est indécidable dans le cas général. Toutefois, si on limite le nombre de paramètres autorisés dans la définition d'une commande à trois, et en évitant les créations cycliques d'objets, le modèle résultant est décidable en temps polynômial, tout en restant suffisamment expressif pour représenter un grand nombre de politiques de sécurité.

1.2.4.2 Modèles basés sur la notion de rôle

La nature de l'organisation ou du système cible de la politique de sécurité peut également fournir des informations permettant de faciliter la définition ou l'utilisation de cette politique. L'objectif des modèles de sécurité basés sur la notion de *rôle* est de tirer parti de la structure de l'organisation considérée, afin de l'intégrer à la description de la politique de sécurité. Les atouts de cette approche sont doubles:

- d'une part, l'organisation elle-même fournit une base de départ à la définition des éléments de la politique de sécurité, et peut suggérer une structure et un certain nombre de règles adaptées à cette organisation;
- d'autre part, la mise en œuvre et l'évolution éventuelle de la politique de sécurité sont facilitées car celle-ci prend en compte, dès sa définition, certains des aspects du fonctionnement de l'organisation, et notamment la manière dont les différentes tâches sont distribuées dans l'organisation.

Un modèle de contrôle d'accès basé sur la notion de rôle n'associe pas directement les différents privilèges (au sens d'ensembles de droits) aux utilisateurs du système. Les privilèges sont associés à une abstraction intermédiaire, appelée un **rôle**. Aux différents utilisateurs sont associés un ou plusieurs rôles et ces deux relations (utilisateur,rôle) et (rôle,privilège) permettent de définir précisément les permissions accordées à un utilisateur particulier. Les rôles peuvent être organisés de manière à former une *hiérarchie de rôles*, qui permet de raffiner progressivement

les différentes permissions attribuées à chaque rôle en structurant la description. Ainsi, si le rôle "chargé de portefeuille bancaire" est inclus dans le rôle "employé de banque", le rôle "chargé de portefeuille bancaire" héritera des privilèges associés au rôle "employé de banque" (comme la possibilité de réaliser des opérations courantes au guichet). Les privilèges issus de l'héritage pourront être complétés par des privilèges spécifiques. Par exemple, le rôle "chargé de portefeuille bancaire" pourra également disposer du privilège d'accorder des prêts à la consommation pour de faibles montants. Enfin, des **contraintes** peuvent être associées au modèle de manière à fournir un moyen d'imposer sur ce modèle un certain nombre de propriétés nécessaires à la réalisation des objectifs de sécurité [Chen & Sandhu 1995]. Par exemple, ces contraintes peuvent impliquer la mise en place de règles obligatoires du type de celles rencontrées dans les politiques multi-niveaux [Sandhu 1996], ou imposer des propriétés sur les rôles eux-mêmes comme l'exclusion mutuelle de deux rôles.

Un modèle de contrôle d'accès basé sur la notion de rôle, ou modèle **RBAC** (*"Role Based Access Control"*), est défini par [Sandhu *et al*. 1996]:

- U, R, P et S, respectivement des ensembles d'utilisateurs, de rôles, de privilèges et de sessions;
- $PA \subseteq R \times P$, une relation associant un privilège à un rôle;
- $UA \subseteq U \times R$, une relation associant un ou plusieurs rôles à un utilisateur;
- $RH \subseteq R \times R$, une hiérarchie de rôles partiellement ordonnée (la relation d'ordre étant notée $\succeq$);
- $user : S \rightarrow U$, une fonction associant chaque session s_i à un seul utilisateur $user(s_i)$, qui reste constant pour toute la durée de vie de la session;
- $roles : S \rightarrow 2^R$, une fonction associant chaque session s_i à un ensemble de rôles $roles(s_i) \subseteq \{r \mid (\exists r' \succeq r)[(user(s_i), r') \in UA]\}$ (qui peut changer pendant la durée de vie de la session), de telle manière que la session s_i soit associée aux privilèges $\bigcup_{r \in roles(s_i)} \{p \mid (\exists r'' / r \succeq r'')[(r'', p) \in PA]\}$;
- et une collection de *contraintes* qui détermine si certains éléments du modèle RBAC sont acceptables (seuls les éléments acceptables étant effectivement intégrés dans le modèle).

Les notions de rôle et de *groupe* peuvent prêter à confusion [Sandhu 1995]. Un groupe est un ensemble d'utilisateurs, et éventuellement d'autres groupes. Un groupe offre donc un moyen de structurer la description des différents éléments présents dans le système, mais ne précise pas les règles de sécurité associées à ces éléments. Par opposition, un rôle définit un ensemble abstrait de privilèges qui pourra être utilisé (un peu à la manière d'une variable dans la définition d'une fonction mathématique) pour définir de manière générique les contraintes de sécurité applicables à différents éléments du système. Ces deux notions s'attachent effectivement à structurer la description du système pour définir la politique de sécurité, mais elles sont différentes, tout en étant compatibles.

L'inclusion dans le modèle RBAC des notions de privilège et de contrainte permet d'adapter cette approche à toutes les politiques de sécurité. En ce sens, le modèle RBAC est neutre, c'est-à-dire indépendant de la politique de sécurité, et peut être utilisé dans le cadre de politiques de sécurité très différentes.

En effet, l'intérêt majeur de la modélisation de la sécurité basée sur les rôles consiste avant tout dans la manière dont elle permet de satisfaire un certain nombre de besoins indépendants des objectifs de sécurité eux-mêmes. Par exemple, l'utilisation des rôles permet de spécifier les privilèges associés à chaque tâche ou fonction présente dans l'organisation, puis indépendamment, d'affecter des utilisateurs précis aux différents rôles. L'intégration de nouveaux utilisateurs, ou même la définition de nouveaux objectifs dans la politique de sécurité en sont grandement facilitées. De même, cette séparation claire entre la structure de la définition de la politique de sécurité et les autorisations effectivement accordées aux individus peut permettre d'améliorer la séparation des pouvoirs au sein de l'organisation. Une contrainte de ce type s'applique en effet directement à l'association des rôles aux privilèges dans la politique et s'étend ensuite aux autorisations des utilisateurs. Enfin, on peut envisager que l'administrateur de sécurité, en charge de la définition de la hiérarchie des rôles, ainsi que des contraintes qui doivent s'y appliquer, puisse cependant déléguer la définition effective du contenu de chaque rôle à d'autres utilisateurs. On peut donc envisager, comme dans [Sandhu *et al*. 1996], de définir des rôles d'administration qui puissent être organisés indépendamment. Ceci facilite évidemment la définition et la mise en œuvre de la politique de sécurité [Jonsher & Gerhardt 1991 ; Lawrence 1993 ; Mohammed & Dilts 1994 ; Solms & Merwe 1994].

1.2.5 Les politiques multi-niveaux

Les politiques d'autorisation basées sur une approche multi-niveaux reposent sur une partition de l'ensemble des sujets et de l'ensemble des objets présents dans le système. Un **niveau** est associé à chaque partition. Les niveaux de sécurité sont généralement partiellement ou totalement ordonnés. Les objectifs de sécurité, qui peuvent être relatifs à la confidentialité ou l'intégrité des objets, sont formulés vis-à-vis de ces différents niveaux, et les règles du schéma d'autorisation qui régissent les contrôles obligatoires prescrits par la politique de sécurité s'appuient également sur ces niveaux.

1.2.5.1 La politique du DoD

La politique obligatoire du DoD, formalisée par Bell et LaPadula [Bell & LaPadula 1975], est une politique d'autorisation multi-niveaux appliquée à la confidentialité. En même temps que la définition de cette politique de sécurité, [Bell & LaPadula 1975] introduit un modèle de sécurité basé sur la notion de treillis qui formalise les objectifs de sécurité et le schéma d'autorisation de leur politique, sur lequel on peut s'appuyer pour en démontrer la cohérence.

Les modèles basés sur la notion de treillis s'appuient sur l'association de différents niveaux aux sujets et aux objets du système. On appelle le niveau $h(s)$ d'un sujet s son niveau d'habilitation, et le niveau $c(o)$ d'un objet o son niveau de classification. Chaque niveau est caractérisé par une **classification** cl, prise dans un ensemble totalement ordonné (par exemple parmi : NON-CLASSIFIÉ, CONFIDENTIEL, SECRET, et TRÈS-SECRET), et par un **compartiment** C défini par un ensemble de catégories (telles que "nucléaire", "défense", "cryptographie", etc.).

La classification cl attribuée à chaque objet est un moyen de représenter le danger que peut constituer la divulgation de cette information. De plus, chaque information est associée à un compartiment C qui identifie les catégories de l'information. Le niveau d'habilitation associé à chaque utilisateur comprend également une classification qui représente la confiance qui lui est accordée, et un compartiment désignant les catégories pour lesquelles cette confiance lui est accordée.

Les niveaux de sécurité constituent un *treillis* partiellement ordonné par une relation de dominance notée $\preceq$. Si $n = (cl, C)$ et $n' = (cl', C')$ sont deux niveaux de sécurité où cl et cl' désignent des classifications et C et C' des compartiments, cette relation est définie par :

$$n \preceq n' \quad \text{si et seulement si} \quad cl \leq cl' \text{et } C \subseteq C'$$

Les *objectifs* de sécurité de cette politique sont :

- d'interdire toute fuite d'information d'un objet possédant une certaine classification vers un objet possédant un niveau de classification inférieur ;
- et d'interdire à tout sujet possédant une certaine habilitation d'obtenir des informations d'un objet d'un niveau de classification supérieur à cette habilitation.

Le schéma d'autorisation associé à ces objectifs de sécurité en découle directement. On considère que l'on peut distinguer vis-à-vis de la confidentialité, parmi les différentes opérations qu'un sujet peut effectuer sur un objet, les opérations de *lecture* et d'*écriture*, et on introduit les *règles* suivantes :

- un sujet ne peut accéder en lecture à un objet que si le niveau d'habilitation du sujet domine le niveau de classification de l'objet ("règle simple") ;
- un sujet ne peut accéder à la fois en lecture à un objet o et en écriture à un objet o' que si le niveau de classification de o' domine le niveau de classification de o ("règle ★").

Dans le modèle de Bell-LaPadula, le système est représenté par une machine à états finis, dont les états sont définis comme une matrice $M \subset (S \times O \rightarrow A)$ qui pour chaque sujet $s \in S$ et chaque objet $o \in O$ décrit les droits d'accès $a \in A$ accordés en lecture ou en écriture à ce sujet sur cet objet (avec donc $A = \{\text{read}, \text{write}\}$). À chaque sujet et à chaque objet sont associés un niveau de sécurité $h(s)$ et $c(o)$, respectivement. Deux propriétés, correspondant aux deux règles de sécurité énoncées précédemment assurent qu'un état est sûr :

- la propriété simple : $\forall s \in S, \forall o \in O$, $\text{read} \in M(s, o) \Rightarrow c(o) \preceq h(s)$
- la propriété ★ :
  $\forall s \in S, \forall (o, o') \in O^2$, $\text{read} \in M(s, o) \wedge \text{write} \in M(s, o') \Rightarrow c(o) \preceq c(o')$

L'utilisation de cette politique de sécurité présente plusieurs inconvénients :

- D'une part, l'information se dégrade constamment par surclassification. En effet, les règles imposées par le schéma d'autorisation font que le niveau de sécurité d'une information ne peut qu'augmenter dans le système, amenant peu à peu ce dernier dans un état où il n'existe que peu de personnes habilitées à les obtenir.
- D'autre part, ce modèle ne représente pas tous les flux d'information et ne permet pas de prendre en compte les canaux cachés (cf page 9) qui peuvent exister dans le système.

La notion de treillis peut servir de support de modélisation des propriétés de sécurité attendues du système pour des politiques de sécurité différentes de celle de Bell et LaPadula. Une application de cette modélisation basée sur la structure de treillis à un certain nombre de politiques de sécurité peut être trouvée dans [Sandhu 1993]. Une extension de cette politique de sécurité visant à l'adapter au contexte des systèmes d'objets distribués a également été proposée dans [Nicomette 1996].

1.2.5.2 La politique d'intégrité de Biba

La politique de sécurité introduite par [Biba 1977] est une politique duale de celle de Bell-LaPadula dans laquelle il s'agit d'assurer l'intégrité des objets présents dans le système. Les niveaux associés aux sujets et aux objets correspondent alors à des **niveaux d'intégrité**. Les *objectifs* de sécurité de cette politique sont donc :

- d'interdire toute propagation d'information d'un objet situé à un certain niveau d'intégrité vers un objet situé à un niveau d'intégrité supérieur ;
- et d'interdire à tout sujet situé à un certain niveau d'intégrité de modifier un objet possédant un niveau d'intégrité supérieur.

Le *schéma d'autorisation* découle de la recherche de ces propriétés, en considérant des labels d'intégrité et le fait que les opérations peuvent être regroupées en trois classes : la *modification* ou l'*observation* d'un objet par un sujet, et l'*invocation* d'un sujet par un autre sujet.

- Un sujet ne peut modifier un objet que si le label d'intégrité du sujet domine le label d'intégrité de l'objet.
- Un sujet ne peut observer un objet que si le label d'intégrité de l'objet domine le label d'intégrité du sujet.
- Un sujet s ne peut invoquer un sujet s' que si le label d'intégrité de s' domine le label d'intégrité de s.

Un inconvénient de cette politique, duale de celle de Bell-LaPadula, réside dans le fait que le niveau d'intégrité de chaque information dans le système ne peut que se dégrader constamment au fur et à mesure de son évolution.

1.2.6 Les politiques de contrôle de flux

Le modèle de Bell-LaPadula ne considère que les flux d'information passant par des objets identifiés (documents, fichiers) et donc ne fournit pas de solution pour l'identification et l'élimination des canaux cachés. En effet, ce type de modèle ne gère en aucun cas les flux d'information atteignant un sujet sans passer par l'intermédiaire d'objets identifiés. Les modèles de contrôle de flux correspondent à une vue plus générale du système. Ils ne considèrent plus seulement des opérations de lecture et écriture sur des objets, mais également des flux d'information entre sujets. Ces modèles s'attachent alors à spécifier les canaux de transmission d'information présents dans le système, à préciser les canaux légitimes et à identifier les canaux cachés.

Une approche originale pour la représentation des flux d'information à l'intérieur d'un système consiste à caractériser les dépendances causales qui existent entre les différents objets présents dans le système à différents instants [d'Ausbourg 1994]. On considère qu'un objet est observable par un utilisateur qui observe une sortie s'il est relié causalement à cette sortie. Dans ce modèle, un système est représenté par un ensemble de points (o, t). Un point désigne l'état d'un objet o à un instant donné t. Certains de ces points sont des *entrées*, d'autres des *sorties* du système, et enfin tous les autres constituent des points *internes*. L'ensemble de ces points évolue avec le temps et cette évolution est due aux transitions élémentaires qui ont lieu dans le système. Une transition élémentaire peut, à un instant t, associer une nouvelle valeur v à un objet o en ce point. Cet instant et cette nouvelle valeur dépendent donc de certains autres points antérieurs.

Cette dépendance fonctionnelle d'un point vis-à-vis de points antérieurs est appelée une **dépendance causale** [Bieber & Cuppens 1992]. La dépendance causale de (o, t) vis-à-vis de (o', t') avec $t' < t$ est notée $(o', t') \rightarrow (o, t)$.

La fermeture transitive de la relation "$\rightarrow$" (notée "$\rightarrow$*") au point (o, t) définit le **cône de causalité** en ce point : $cone(o, t) = \{(o', t') | (o', t') \rightarrow^* (o, t)\}$.

Réciproquement, on définit le **cône de dépendance** comme l'ensemble des points qui dépendent causalement de (o, t) : $dep(o, t) = \{(o', t') | (o, t) \rightarrow^* (o', t')\}$.

Ces dépendances causales représentent la structure des flux d'information dans le système. Si un sujet s possède une certaine connaissance du comportement interne du système, il est en mesure de connaître cette structure interne des dépendances causales. Dans ce cas, en observant une sortie particulière x_o, il peut être en mesure d'inférer toute information appartenant à $cone(x_o)$. Réciproquement, en altérant une entrée x_i du système, s peut éventuellement altérer tous les points appartenant à $dep(x_i)$.

Les objectifs de sécurité qu'il est possible de définir dans ce modèle peuvent être relatifs à la confidentialité ou à l'intégrité du système. En particulier, si un sujet s peut observer un ensemble O_s de sorties x_o, du système, on note Obs_s l'ensemble des points que s peut observer dans le système:

$$Obs_s = \bigcup_{x_o \in O_s} cone(x_o)$$

De manière identique, si un sujet s peut modifier un ensemble A_s d'entrées x_i du système, on note Alt_s l'ensemble des points que s peut modifier:

$$Alt_s = \bigcup_{x_i \in A_s} dep(x_i)$$

Si R_s est l'ensemble des points que le sujet s a le droit d'observer d'après la politique de sécurité du système, on peut dire que le système est sûr (vis-à-vis de la confidentialité) si le sujet s ne peut observer que les objets qu'il a le doit d'observer, c'est-à-dire si: $Obs_s \subseteq R_s$. Si W_s est l'ensemble des points que le sujet s a le droit de modifier d'après la politique de sécurité du système, on peut dire de manière similaire que le système est sûr (vis-à-vis de l'intégrité) si le sujet s ne peut agir que sur les objets qu'il a le droit de modifier, c'est-à-dire si: $Alt_s \subseteq W_s$.

En considérant un ensemble de niveaux associés aux sujets et aux objets, la propriété $Obs_s \subseteq R_s$ relative à la confidentialité peut être obtenue en imposant deux *règles* dans le système analogues à celles définies dans la politique de Bell-LaPadula [d'Ausbourg 1994]:

- un sujet n'est autorisé à observer que des objets dont la classification est dominée par son habilitation;
- et, si un objet o' dépend causalement d'un objet o, alors la classification de o' doit dominer la classification de o.

Ce modèle est particulièrement intéressant parce qu'il introduit une nouvelle manière de formaliser les flux d'information dans un système. L'intérêt principal de cette formalisation réside dans son aspect minimal: la notion de dépendance causale permet de décrire de manière très stricte un flux d'information. Toutefois, les implémentations de ce modèle qui ont été réalisées semblent limitées à des applications assez spécifiques [Calas 1995].

1.2.7 Les politiques de contrôle d'interface

Plutôt que de spécifier des mécanismes particuliers permettant d'assurer la sécurité, les politiques de contrôle d'interface spécifient des restrictions sur les entrées et les sorties du système qui permettent d'obtenir des propriétés de sécurité. Ce type de politique de sécurité s'intéresse plus directement au comportement dynamique du système et s'appuie sur des méthodes de modélisation très générales. Elles considèrent une représentation du système incluant les différents sujets (ou utilisateurs) et l'ensemble des *traces d'exécution* associées à ces utilisateurs. Une trace est définie comme l'historique des entrées, c'est-à dire la suite ordonnée de tous les états successifs du système entre chaque entrée (ou transition ou commande). Certaines

commandes particulières déterminent les sorties du système effectuées par un utilisateur, et on s'intéresse principalement aux propriétés que le système possède vis-à-vis de toutes ses sorties.

La principale qualité de cette approche très abstraite est d'avoir permis des avancées significatives dans la compréhension des problèmes de formalisation posés par la sécurité. Un certain nombre de propriétés importantes ont pu être étudiées et comparées en se basant sur ces modélisations très générales des systèmes.

Les principales propriétés identifiées dans la littérature, et qui constituent les différents *objectifs* de sécurité qui peuvent être choisis dans ces politiques de sécurité sont:

- La **non-interférence**, au sens où un groupe d'utilisateurs, utilisant un certain ensemble de commandes, n'interfère pas avec un autre groupe d'utilisateurs si ce que fait le premier groupe avec ces commandes n'a aucun effet sur ce que le deuxième groupe d'utilisateurs peut observer [Goguen & Meseguer 1982]. Étant donné le modèle dans lequel elle est définie formellement, cette propriété s'applique aux systèmes *déterministes*.
- La **non-déductibilité**, correspondant au fait que, quelle que soit la sortie observée par un utilisateur possédant une habilitation de bas niveau, cette sortie est compatible avec n'importe quelle entrée acceptable d'un utilisateur possédant une habilitation de haut niveau[1]. Cette propriété peut s'appliquer aux systèmes *non déterministes*.
- La **non-interférence généralisée**, qui correspond à un renforcement de la non-déductibilité pour parer à un problème identifié initialement par [McCullough 1987]: il fait remarquer que la propriété de non-déductibilité ne garantit pas qu'un utilisateur de bas niveau n'ait pas directement accès à des informations de haut niveau, à condition qu'elles soient mêlées à des données aléatoires[2].
- La **restriction**, ou non-inférence, qui restreint encore la propriété de non-interférence généralisée afin d'obtenir une propriété composable à partir des propriétés de plusieurs sous-systèmes [McCullough 1990] pour des systèmes *non déterministes*.

Les modèles d'interface considèrent donc une représentation du système sous la forme d'un automate à états finis dont les sorties sont observables. Un tel système est constitué par:

- un ensemble S de sujets ou utilisateurs;
- un ensemble Σ d'états du système;

1. Ce qui signifie qu'un utilisateur ayant une habilitation de bas niveau ne peut pas déduire, à partir des sorties qu'il observe, quelles entrées ont été effectuées par un utilisateur de haut niveau.

2. En fait, dans un système non déterministe, on peut considérer que la propriété de non-déductibilité ne permet pas de faire la différence entre un bruit aléatoire mêlé à de l'information, et un véritable cryptogramme.

- un ensemble Γ de commandes ou opérations pouvant être effectuées sur le système;
- un ensemble Out dont les éléments sont les "sorties" visibles par un utilisateur;

ainsi que:

- une fonction $out: \Sigma \times S \rightarrow \text{Out}$ qui représente ce qu'un utilisateur donné observe quand la machine est dans un certain état, appelée la *fonction de sortie*;
- une fonction $do: \Sigma \times S \times \Gamma \rightarrow \Sigma$ qui représente la manière dont les états sont transformés par les commandes, appelée la *fonction de transition*;
- et la constante $\sigma_0 \in \Sigma$, état initial de la machine.

Si w est une chaîne d'entrée ou **trace** de ce système, c'est-à-dire une suite ordonnée de commandes de Γ effectuées par les utilisateurs $w \in traces$ avec $traces = (S \times \Gamma)^*$, on note $[w]$ l'état atteint par la machine à état après application par les utilisateurs indiqués de toutes les commandes représentées dans w, en partant de l'état initial σ_0. On notera $\langle\ \rangle$ la trace vide (ne comprenant aucune commande), et, in extenso, $v \cdot \gamma_1(u_1) \cdot \gamma_2(u_2) \cdot \ldots \cdot \gamma_n(u_n)$ la trace w constituée par la trace v (éventuellement vide) suivie par la séquence de commandes $(\gamma_i)_{1 \le i \le n}$ de Γ effectuées par les utilisateurs $(u_i)_{1 \le i \le n}$.

Cette machine à états peut être naturellement étendue pour prendre en compte une notion de sécurité multi-niveaux telle que celle de Bell-LaPadula, constituant ainsi un système avec capacités [Goguen & Meseguer 1982]. Il suffit dans ce cas de considérer un espace d'état incluant une matrice de contrôle d'accès et des commandes de modification des droits d'accès telles que définies au 1.2.5.1. Il est également d'usage d'isoler les commandes de Γ qui permettent d'effectuer des entrées ou des sorties vers l'utilisateur et de considérer des traces constituées par une série d'entrées dans le système (c'est-à-dire de commandes) et terminées par une opération de sortie. L'ensemble des opérations de sortie est noté Γ_{out}. C'est en effet à ces opérations particulières qu'une politique de sécurité visant à assurer des propriétés de confidentialité s'intéressera principalement. À chaque fois qu'on effectuera cette distinction dans la description des propriétés suivantes, les noms des commandes, tels que $\text{read}(u)$, $\text{highin}(u)$, $\text{lowout}(u)$, $\text{lowin}(u)$, indiquent sans ambiguïté leur catégorie.

1.2.7.1 Systèmes déterministes: Non-interférence

Soit h une fonction définissant les habilitations des utilisateurs, telle que $h(u)$ soit l'habilitation de u (cf 1.2.5.1). Soit *purge* une fonction de $S \times traces$ dans S telle que:

$$purge(u, \langle\ \rangle) = \langle\ \rangle$$

$$purge(u, hist \cdot command(u')) = \begin{cases} purge(u, hist) \cdot command(u') & \text{si } h(u) \succeq h(u') \\ purge(u, hist) & \text{si } h(u) \prec h(u') \end{cases}$$

Un système satisfait la propriété de **non-interférence** si et seulement si:

$$\forall u \in S, \forall w \in traces, \forall c \in \Gamma_{out}$$
$$out(u, w \cdot c(u))) = out(u, purge(u, w) \cdot c(u))$$

Il n'est pas toujours aisé de comparer exactement le modèle de Bell-LaPadula et le modèle basé sur la non-interférence [McLean 1990]. Néanmoins, on peut noter qu'en général le modèle de Bell-LaPadula fournit des propriétés plus faibles que la non-interférence dans le sens où cette dernière interdit la plupart des canaux cachés qui seraient utilisables dans l'interprétation standard des primitives du modèle de Bell-LaPadula [Millen 1987]. D'un autre coté, la propriété de non-interférence autorise l'implémentation d'opérations qui ne seraient pas permises dans un système basé sur le modèle de Bell-LaPadula, comme la possibilité pour un utilisateur habilité à un bas niveau de confidentialité de copier directement un fichier classifié à un haut niveau dans un autre fichier classifié au même niveau (sans que l'utilisateur y accède lui-même). Dans les deux cas toutefois, la propriété de non-interférence semble mieux capturer la notion intuitive de confidentialité que le modèle de Bell-LaPadula.

Ce modèle souffre pourtant d'un certain nombre de limitations. D'une part, la propriété de non-interférence est extrêmement forte et peut même être considérée comme trop forte car, par exemple, elle conduit à interdire l'usage de canaux de communications cryptés (même parfaits) entre utilisateurs de haut niveaux si des utilisateurs de bas niveaux peuvent avoir accès au cryptogramme. D'autre part, ce modèle s'applique uniquement aux systèmes déterministes. En dépit de ses limitations et des difficultés d'application qu'il représente, le modèle de non-interférence constitue l'état de l'art actuel pour les modèles de systèmes déterministes. Son extension en direction des systèmes non déterministes a fait l'objet de nombreux travaux.

1.2.7.2 Systèmes non déterministes: Non-déductibilité, Non-interférence Généralisée, Restriction

Afin de donner une version non déterministe de la propriété de non-interférence, il s'agit d'abord de présenter la manière dont on peut décrire un système non déterministe. Si on reprend la définition précédente, on peut considérer qu'une trace d'exécution représente un comportement *acceptable* du système. Dans ce cas, un système non déterministe est décrit par un *ensemble* de traces (acceptables). Afin de définir les propriétés suivantes, on distingue également l'existence de deux *niveaux* d'interaction avec le système (au sens de Bell-LaPadula) qui correspondent à un *haut niveau* et à un *bas niveau* de confidentialité.

La propriété de **non-déductibilité**, proposée par [Sutherland 1986], correspond au fait que pour toute paire de traces acceptables T et T', il existe une trace acceptable T'' qui contient: les commandes de bas niveau de sécurité de T (dans leur ordre respectif), les entrées de haut niveau de T' (dans leur ordre respectif), et éventuelle-

ment d'autres commandes distinctes de celles-ci. Cette propriété correspond au fait que tout ce qu'un utilisateur d'un bas niveau d'habilitation observe est compatible avec n'importe quelle entrée acceptable d'un utilisateur de haut-niveau.

Bien que la propriété de non-déductibilité pour un système soit plus générale que la propriété de non-interférence dans le sens où elle ne suppose pas que le système soit déterministe, elle n'est pas équivalente à la non-interférence pour des systèmes déterministes comptant plus de deux utilisateurs. En fait la non-déductibilité est une propriété plus faible que la non-interférence dans ce cas[1].

Cette propriété est cependant trop faible, et pose un problème identifié dans [McCullough 1987], qui a conduit à l'introduction d'une version corrigée, appelée la propriété de non-interférence généralisée. Un système possède la propriété de **non-interférence généralisée** si et seulement si, étant donnée une trace acceptable T pour le système, et une trace altérée T' construite en insérant ou en supprimant une entrée de haut niveau de T, il existe une trace acceptable T'' construite en insérant ou en supprimant une sortie de haut niveau de T' juste après l'altération de T ayant conduit à T' [McCullough 1987].

Les propriétés de non-déductibilité et de non-interférence généralisée souffrent cependant toutes deux d'un inconvénient majeur: ce ne sont pas des propriétés préservées par la composition de deux systèmes [McCullough 1987 ; McCullough 1990; Zakinthinos & Lee 1994]. La propriété de restriction a été introduite dans le but d'apporter une solution à ce problème.

Un système possède la propriété de **restriction** si et seulement si, étant donnée une trace acceptable T pour le système, et une trace altérée T' construite en insérant ou en supprimant une entrée de haut niveau de T, il existe une trace acceptable T'' construite en insérant ou en supprimant une sortie de haut niveau de T', juste après l'altération de T ayant conduit à T', et après chaque séquence d'entrées de bas niveau qui suivent immédiatement l'altération de T [McCullough 1990].

1.2.8 Les politiques et modèles spécifiques

1.2.8.1 La politique de Clark et Wilson

La politique de sécurité dédiée à l'intégrité définie par Clark et Wilson [Clark & Wilson 1987] s'intéresse aux contraintes de sécurité définies dans le contexte d'une organisation commerciale. Ce souci de l'environnement dans lequel la politique de sécurité est mise en œuvre provient de la constatation que des politiques de sécurité comme les politiques multi-niveaux ne sont généralement appliquées directement que dans des organisations très rigides, telles que les

1. Dans le cas d'un système déterministe avec exactement deux utilisateurs, les propriétés de non-interférence et de non-déductibilité sont équivalentes [Bieber & Cuppens 1992].

organisations militaires. Dans le cadre d'une organisation commerciale, d'autres approches sont mises en œuvre pour obtenir des propriétés de sécurité, notamment en ce qui concerne l'intégrité des données.

La politique de sécurité proposée par Clark et Wilson repose d'abord sur la séparation des données manipulées en deux groupes : les **données contraintes** qui sont soumises à des règles de manipulation strictes visant à garantir leur intégrité, et les **données non-contraintes** qui doivent faire l'objet d'une vérification avant d'être utilisées par le système, comme les entrées fournies par les utilisateurs par exemple. Les différentes opérations de transformation des données pouvant être effectuées par le système doivent permettre de garantir que l'intégrité des données contraintes persiste. Ceci implique donc une certification des opérations de transformation qui sont autorisées à manipuler des données contraintes, ainsi que l'existence d'opérations de validation des données permettant de vérifier leur intégrité. Par ailleurs, les opérations acceptant comme paramètres d'entrée des données non-contraintes doivent garantir que le résultat de la transformation produit une donnée dont l'intégrité a été vérifiée.

Afin de représenter les règles appliquées dans leur politique de sécurité, Clark et Wilson définissent [Clark & Wilson 1987] :

- l'ensemble U des utilisateurs du système, dont les éléments sont notés u_1, u_2,…
- l'ensemble CDI des données contraintes (*Constrained Data Item*), dont les éléments sont notés CDI_1, CDI_2,…
- l'ensemble UDI des données non-contraintes (*Unconstrained Data Item*), dont les éléments sont notés UDI_1, UDI_2,…
- l'ensemble TP des opérations de transformation des données (*Transformation Procedure*), dont les éléments sont notés TP_1, TP_2,…
- un ensemble IVP d'opérations de vérification de l'intégrité des données (*Integrity Verification Procedure*), dont les éléments sont notés IVP_1, IVP_2,…
- une relation R_c liant chaque opération de transformation TP_i à un sous-ensemble de CDI, qui précise les données que l'opération peut manipuler.
- une relation R_u liant chaque utilisateur u_i et chaque opération de transformation TP_i a un sous-ensemble de CDI, qui précise les données contraintes qu'un utilisateur est autorisé à manipuler par le biais de TP_i.

L'*objectif* de cette politique de sécurité est donc de garantir l'intégrité d'un certain nombre de données clairement identifiées. Afin de satisfaire également un besoin de séparation des pouvoirs entre les différents utilisateurs présents dans l'organisation, les différentes opérations de transformation qui visent des données particulières doivent également être explicitement autorisées pour chaque utilisateur.

Les règles obligatoires imposées dans le modèle de Clark et Wilson pour mettre en œuvre la politique d'intégrité sont:

- Les opérations de IVP doivent assurer que toutes les données de CDI sont dans un état valide (du point de vue de l'intégrité) au moment où les opérations de IVP sont exécutées.
- Toutes les opérations de TP doivent être certifiées, c'est-à-dire que, pour chaque donnée contrainte associée à une opération par R_c, l'opération doit garantir que, si une donnée contrainte est valide avant l'exécution de l'opération, celle-ci est valide après l'exécution de l'opération.
- R_u doit être certifiée afin de refléter les besoins de séparation de pouvoirs de l'organisation.
- Le système doit garantir que chaque opération de TP n'est effectuée que sur les données contraintes spécifiée par R_c.
- Le système doit garantir que chaque utilisateur n'effectue des opérations de transformation sur des données contraintes que si celles-ci lui sont associées par R_u.
- Le système doit authentifier l'identité de chaque utilisateur souhaitant exécuter une opération de TP.
- Toutes les opérations de TP doivent inscrire dans une donnée contrainte particulière toutes les informations nécessaires pour identifier l'opération qui a été effectuée (audit).
- Enfin, chaque opération TP_i de TP qui accepte une donnée UDI_i de UDI en entrée doit garantir que, quelle que soit la valeur de UDI_i, soit TP_i n'effectue que des transformations conduisant à une donnée contrainte valide CDI_i, soit UDI_i est rejetée.

En dépit de sa présentation relativement informelle, le modèle de Clark et Wilson met clairement en évidence un certain nombre de préoccupations qui peuvent apparaître dans le contexte d'une organisation commerciale. Tout d'abord, ce modèle s'intéresse à l'assurance de propriétés d'intégrité par le biais de l'utilisation de procédures de transformation certifiées. En pratique, ceci limite bien évidemment le nombre d'opérations pouvant être définies dans le système, et surtout l'évolution de ces opérations. Néanmoins, c'est une approche qui offre l'avantage de faciliter le fonctionnement et la conception du système en concentrant la majeure partie de l'effort de sécurité au niveau d'une activité indépendante de certification. Ensuite, ce modèle met en évidence deux préoccupations importantes dans le cadre d'une organisation commerciale: la *traçabilité* des opérations, c'est-à-dire la possibilité de reconstituer toutes les actions importantes (du point de vue des objectifs de sécurité) effectuées par le système; et la *séparation des pouvoirs* qui reste un facteur important de sécurité prenant en compte les utilisateurs.

Enfin, ce modèle de sécurité admet, quoique de manière un peu implicite, la possibilité que le système dévie de son fonctionnement normal. En effet, si les propriétés d'intégrité ne sont pas satisfaites à un moment donné, l'existence de procédures de validation de l'intégrité, ajoutée au fait que les entrées non-contraintes sont acceptées par certaines procédures de transformation, offre des moyens de détection

d'une infraction aux objectifs de sécurité et de retour vers un état satisfaisant. L'existence d'un historique de l'exécution du système (imposé par la politique de sécurité) permet alors de reconstituer l'histoire du système et d'identifier les comportements erronés qui ont pu être à l'origine d'une violation des objectifs de sécurité (par exemple une faute dans l'implémentation d'une opération, non-détectée par la certification). Ce souci de fournir, outre des mécanismes garantissant la sécurité du système, des mécanismes capables de fonctionner en l'absence des propriétés de sécurité attendues pour le système, participe à la volonté de définir des politiques de sécurité applicables dans un environnement moins rigide que ceux dans lesquels des politiques de sécurité comme les politiques multi-niveaux sont utilisées.

1.2.8.2 La politique de la muraille de Chine

La politique de sécurité dite de "la muraille de Chine", présentée dans [Brewer & Nash 1989], est une politique de sécurité spécifique issue des règlements de sécurité ayant cours dans les institutions financières britanniques. Dans ce type d'organisation, un souci très particulier est accordé au cloisonnement des différentes informations pouvant être relatives à des clients concurrents. Dans le cas où un organisme financier est amené à traiter des opérations pour le compte de deux sociétés concurrentes, les personnels de cet organisme ne sont autorisés à accéder qu'aux informations concernant l'une des deux sociétés. Une fois des informations connues concernant l'une d'elles, tout accès aux informations concernant l'autre doit être interdit. Chaque classe d'information est donc obligatoirement séparée de l'autre par une barrière infranchissable une fois le choix initial effectué (un "mur", d'où le nom donné à cette politique).

La formalisation de la politique de la muraille de Chine implique de regrouper les informations contenues dans le système en différents ensembles E_c relatifs à chacune des organisations extérieures avec lesquelles l'organisation cible de la politique de sécurité est en contact. Ces différents ensembles de données sont répartis dans des **classes de conflit d'intérêt** disjointes : COI_1, COI_2,…, COI_n, comptant chacune m_j ensembles de données. On pose :

$$COI_j = \left\{ E_1^{COI_j}, E_2^{COI_j}, \ldots, E_{m_j}^{COI_j} \right\}$$

en notant $E_k^{COI_j}$ l'ensemble des données relatives à la *k*ième organisation, appartenant à la classe de conflit d'intérêt COI_j.

L'*objectif* de cette politique de sécurité est donc de garantir qu'aucun utilisateur n'accède simultanément à des données appartenant à des ensembles en conflit d'intérêt. Ceci est obtenu en imposant deux *règles* de fonctionnement triviales : l'accès à une donnée n'est autorisée que si cette donnée appartient à un ensemble auquel l'utilisateur a déjà accédé, ou si cette donnée appartient à un ensemble qui n'est en conflit avec aucun autre ensemble de données auquel l'utilisateur a préalablement accédé.

Un utilisateur désirant accéder à une donnée d'un ensemble $E_c \in COI_j$ n'est donc autorisé à y accéder que s'il n'existe pas d'ensemble $COI_{i \neq j}$ contenant des données auxquelles il aurait déjà accédé.

Différents contrôles supplémentaires peuvent éventuellement être associés à une politique de sécurité du type de la muraille de Chine, de manière à contrôler l'accès aux différentes données appartenant à un même ensemble E_c (par exemple des contrôles multi-niveaux).

L'intérêt de cette politique de sécurité est de mettre en évidence un besoin de sécurité réel, qui apparaît même dans la législation britannique, mais qui ne correspond pas aux objectifs des politiques de sécurité classiques et notamment des politiques multi-niveaux.

1.2.9 Limites de ces approches

Les différentes politiques de sécurité ainsi que les modèles éventuellement associés que nous venons de détailler peuvent s'appliquer avec plus ou moins de succès. Dans ce paragraphe, nous nous intéressons aux qualités et aux faiblesses de ces différentes approches selon deux points de vue. D'abord, nous examinons l'impact de l'utilisation de ces méthodologies dans le contexte d'un système d'information quelconque (et pas seulement informatique). Enfin, du point de vue d'un système informatique, pour lequel elles ont souvent été initialement développées, nous identifions les problèmes que ces approches peuvent poser.

1.2.9.1 Dans le cadre d'un système d'information

Ainsi que nous avons pu le voir, la plupart des modèles de sécurité présentés dans les sections précédentes introduisent des règles de contrôle obligatoires basées sur la définition de niveaux. Ceci montre le rôle majeur joué par les politiques multi-niveaux dans les représentations de la sécurité étudiées jusqu'à présent. On retrouve donc souvent la définition de différents niveaux de confidentialité ou d'intégrité dans l'utilisation qui a pu être faite des modèles de sécurité dans les organisations.

Pourtant, dans une organisation, l'introduction de classifications ou de niveaux d'intégrité pour les données, d'habilitations ou de niveaux d'intégrité pour les individus, et des règles obligatoires associées, pose un problème majeur. En effet, elle induit une grande rigidité. Une politique de sécurité telle que la politique de confidentialité de Bell-LaPadula restreint considérablement les flux d'information autorisés et, dans tous les cas, impose des contraintes sur la propagation de l'information. Or, dans beaucoup d'organisations, la souplesse des flux d'information entre individus et l'adaptabilité de l'ensemble de la structure sont généralement considérées comme des atouts. Il en ressort que peu d'organisations peuvent accepter de sacrifier leur flexibilité pour mettre en place une telle politique de sécurité, même si elles présentent des besoins de confidentialité. C'est la principale raison qui fait que les politiques multi-niveaux sont essentiellement utilisées dans des

organisations militaires ou gouvernementales, une entreprise ne pouvant généralement pas tolérer leur mise en place généralisée sans compromettre sa propre efficacité.

Par ailleurs, certains des besoins de sécurité présents dans ces organisations, comme la séparation des pouvoirs, ou les conflits d'intérêt, ne sont pas directement pris en compte par une politique multi-niveaux. Ce sont ces besoins qui ont conduit à la définition de politiques de sécurité spécifiques comme la politique de Clark et Wilson (cf 1.2.8.1) et la politique de la "muraille de Chine" (cf 1.2.8.2). Ces travaux ont permis d'identifier un certain nombre de préoccupations associées à certains systèmes d'information, et qui donnent parfois l'impression qu'une politique de sécurité, dans ce contexte, doit avant tout être spécifique de l'organisation ou du système auquel elle est censée s'appliquer ([McLean *et al*. 1984; McLean 1987]). Il importe d'ailleurs, en règle générale, de pouvoir adapter la politique de sécurité à l'organisation considérée, plutôt que de devoir adapter l'organisation à la politique de sécurité choisie comme c'est généralement nécessaire avec les politiques de sécurité que nous avons présentées.

La possibilité d'utiliser les informations disponibles sur la structure du système ou de l'organisation comme support à la définition de sa politique de sécurité constituent des atouts majeurs qui soutiennent les modèles de sécurité basés sur la notion de rôle (cf 1.2.4.2). En effet, si la notion de rôle présente également un intérêt théorique en proposant l'utilisation d'une forme de paramétrage pour la définition des objectifs de sécurité du système, elle présente avant tout un intérêt pratique: elle reflète le fonctionnement du système d'information considéré, et elle permet d'éclaircir la définition et l'évolution éventuelle de la politique de sécurité. Toutefois, cette approche reste, à l'heure actuelle, indépendante du choix d'une politique de sécurité particulière. Malgré les améliorations apportées par les modèles de sécurité basés sur les rôles, la proposition d'une méthodologie de définition d'une politique de sécurité suffisamment souple pour être adaptable à différents types d'organisations reste un problème ouvert.

1.2.9.2 Dans le cadre d'un système informatique

Quand on considère leur application à un système informatique, les faiblesses des différentes approches étudiées jusqu'à présent sont de nature assez différentes et dépendent de la politique choisie.

Le problème le plus délicat associé à l'utilisation de la politique de Bell-LaPadula dans un système informatique réside dans la nécessité de vérifier l'absence (ou la très faible capacité) de tous les canaux cachés présents dans le système. Le partage de ressources et la complexité des systèmes informatiques font que les moyens de transmission de l'information par des voies non-conventionnelles sont très nombreux. Leur identification exhaustive et leur quantification, pourtant indispensable pour la certification du système [TCSEC 1985], sont des problèmes extrêmement complexes, qui retardent considérablement l'introduction de systèmes sûrs basés sur cette approche. À cette validation difficile s'ajoute bien évidemment le pro-

blème de surclassification progressive de l'information, inhérent au schéma d'autorisation de Bell-LaPadula (ou de dégradation dans le cas de celui de Biba), et qui impose l'introduction de mécanismes de déclassification (ou de vérification d'intégrité) permettant de le résoudre.

Les modèles basés sur le contrôle des interfaces (cf 1.2.7) permettent de choisir des objectifs de sécurité dont les propriétés sont suffisantes pour garantir le contrôle des canaux cachés. Néanmoins, la définition de méthodes de conception de systèmes informatiques correspondant à de tels modèles, et la vérification de l'assurance de propriétés telles que la non-interférence ou la restriction constituent encore le plus souvent des problèmes ouverts [McLean 1994]. Outre l'approche présentée dans [Haigh & Young 1986] s'agissant de la non-interférence, une des rares implémentations basées sur un modèle formel suffisamment puissant pour éviter la recherche et l'élimination spécifique des canaux cachés est basée sur le modèle de causalité (cf 1.2.6) [Calas 1994; Calas 1995]. Toutefois, il semble que cette implémentation ne puisse être utilisée qu'avec un nombre limité de niveaux.

Les modèles généraux basés sur une matrice de contrôle d'accès, présentés au 1.2.4.1, sont indépendants du choix d'une politique de sécurité particulière. Afin de permettre la vérification de propriétés sur le modèle (c'est-à-dire la détermination des états maximaux de la matrice de contrôle d'accès), ils imposent des contraintes particulières sur le schéma d'autorisation, mais celles-ci restent acceptables (tout au moins dans le cas de TAM). Néanmoins, pour un système réel, de telles vérifications sont généralement coûteuses et incomplètes: on ne peut vérifier, sans faire une énumération totale, qu'il n'est pas possible d'atteindre un état d'insécurité. Par ailleurs, l'extension de ces modèles et surtout de leur implémentation vers les systèmes distribués pose de nombreux problèmes. Par exemple, la gestion centralisée d'une matrice de contrôle d'accès couvrant l'ensemble des sites du système a un impact extrêmement négatif sur les performances. Dans la plupart des implémentations, la défaillance d'un des sites met en péril la sécurité de l'ensemble du système. L'adaptation des solutions utilisées pour assurer la sécurité d'un système centralisé à des environnements distribués demande ainsi des travaux spécifiques [TNI 1987; Deswarte *et al.* 1991; Kohl & Neuman 1993]. Ces difficultés peuvent nécessiter l'introduction de concepts nouveaux, encore non-pris en compte formellement dans les modèles de sécurité, comme la délégation de droit mise en œuvre grâce à des coupons dans [Nicomette 1996; Nicomette & Deswarte 1997].

Enfin, dans toutes ces approches, il n'est pas possible d'adapter ces politiques de sécurité pour représenter des propriétés de sécurité spécifiques, recherchées pour un système donné à un instant donné. De ce fait, les différents contrôles obligatoires mis en place sur le système sont généralement mal supportés par les utilisateurs qui les jugent inadaptés et trop stricts. Consciemment ou non, ceux-ci introduisent alors des *vulnérabilités* dans le système en détournant les mécanismes de sécurité. Ce problème correspond tout d'abord à une lacune des modèles de sécurité. Ils sont rarement configurables (à l'exception de ceux introduisant une notion de rôle), et ne permettent pas de faire évoluer les objectifs de la politique de sécurité ni de les

adapter à l'organisation. On peut également considérer que la présence de vulnérabilités dans un système est une situation inévitable, qui ne constitue pas en tant que telle une menace rédhibitoire pour la sécurité globale, mais qui peut dans certains cas la compromettre plus ou moins. La politique de sécurité du système et le modèle qui la reflète devraient donc intégrer la présence de ces vulnérabilités, en fournissant un moyen de les identifier et de déterminer celles qui sont effectivement dangereuses pour la sécurité.

1.3 Évaluation de la sécurité

Suite à la présentation des différentes propriétés de sécurité que l'on peut désirer voir assurer par un système, nous pouvons à présent nous intéresser à la manière dont les systèmes qui implémentent ces propriétés peuvent être évalués. L'évaluation de la sécurité d'un système peut prendre plusieurs formes. Pour les systèmes informatiques on rencontre une approche d'évaluation ordinale par le biais de *critères d'évaluation* normalisés. Dans le cadre des organisations, l'évaluation de la sécurité peut s'appuyer sur des méthodes plus générales, regroupées sous le qualificatif d'*analyse des risques*. Enfin, certaines méthodes d'*évaluation quantitative* de la sécurité ont été proposées, et constituent une voie prometteuse.

1.3.1 Critères d'évaluation

Pour évaluer la capacité des systèmes informatiques à faire face à des malveillances, on utilise généralement des critères d'évaluation normalisés.

1.3.1.1 Le livre orange (TCSEC)

Les premiers critères d'évaluation de la sécurité ont été définis par le *Department of Defense* (DoD) des Etats-Unis dans ce qui est couramment appelé le *Livre Orange* ou TCSEC (*Trusted Computer System Evaluation Criteria*) [TCSEC 1985], ou dans les différents interprétations associées à des livres de diverses couleurs qui l'accompagnent, comme le *Livre Rouge* ou TNI (*Trusted Network Interpretation of the TCSEC*) [TNI 1987]. Ce document a longtemps été la référence en matière d'évaluation de la sécurité des systèmes informatiques. Ces critères, basés à la fois sur des listes de fonctions de sécurité à remplir et sur les techniques employées pour la vérification, conduisent à classer les systèmes en 7 catégories (D, C1, C2, B1, B2, B3, A1 dans un ordre croissant de sécurité).

Quatre familles de critères sont définies pour chaque niveau, traitant respectivement de la *politique d'autorisation*, de l'*audit*, de l'*assurance* et de la *documentation*. La politique d'autorisation stipule une politique précise à suivre (discrétionnaire ou obligatoire) en fonction des différents niveaux de certification visés. La politique obligatoire imposée par le livre orange est celle définie par Bell-LaPadula [Bell & LaPadula 1975]. Le critère d'audit précise les fonctions requises en matière d'identification, d'authentification et d'enregistrement des actions des utilisateurs.

Le critère d'assurance fixe des recommandations concernant les méthodes de conception et de vérification utilisées afin d'augmenter la confiance de l'évaluateur en ce qui concerne le fait que le système implémente bien les fonctionnalités qu'il prétend avoir. Par exemple, le critère d'assurance peut imposer la réalisation d'une vérification formelle de l'implémentation. Le critère de documentation spécifie l'ensemble des documents qui doivent être fournis avec le produit lors de l'évaluation.

Schématiquement, on peut identifier les caractéristiques principales des différents niveaux définis par le livre orange:

- Un système classé au niveau D est un système qui n'a pas été évalué au niveau visé par l'évaluation.
- Jusqu'aux niveaux C1 et C2, un système peut utiliser une politique d'autorisation discrétionnaire.
- Les critères imposent l'utilisation d'une politique obligatoire et d'une politique discrétionnaire pour les niveaux B1, B2 et B3.
- Un système classé au niveau A1 est fonctionnellement équivalent à un système classé au niveau B3, mais ce système est caractérisé par l'utilisation de méthodes formelles de vérification pour assurer que les contrôles discrétionnaires et obligatoires utilisés permettent bien d'assurer la protection des informations sensibles manipulées par le système. Un exemple de système A1 est présenté dans [Weissman 1992].

Pendant des années, les critères du livre orange ont été la seule référence en matière d'évaluation de la sécurité des systèmes informatiques. Pourtant, ces critères visent d'abord à satisfaire les besoins du DoD, c'est-à-dire qu'ils privilégient la confidentialité plutôt que l'intégrité. Par ailleurs, la politique de sécurité choisie (celle de Bell-LaPadula) ne rallie pas tous les suffrages (cf 1.2.9). Enfin, le manque de souplesse et la difficulté de mise en œuvre des critères du livre orange ont initié le développement, à l'extérieur ou à l'intérieur même des Etats-Unis [Federal Criteria 1992], de nouvelles générations de critères. Nous abordons ci-après l'exemple des critères adoptés par la Communauté Européenne, mais d'autres pays, tels que le Canada [CTCPEC 1993] et le Japon [JCSEC 1992] ont également élaboré leur propres critères d'évaluation.

1.3.1.2 Les ITSEC

Les ITSEC sont le résultat de l'harmonisation de travaux réalisés au sein de quatre pays européens: l'Allemagne, la France, les Pays-Bas et le Royaume-Uni [ITSEC 1991]. La différence essentielle que l'on peut noter entre le livre orange et les ITSEC est la distinction entre fonctionnalité et assurance. Une **classe de fonctionnalité** décrit les mécanismes que doit mettre en œuvre un système pour être évalué à ce niveau de fonctionnalité. Une **classe d'assurance** permet, elle, de décrire l'ensemble des preuves qu'un système doit apporter pour montrer qu'il implémente réellement les fonctionnalités qu'il prétend assurer.

Les ITSEC introduisent également la notion de “cible d’évaluation” (TOE pour *Target Of Evaluation*) qui rassemble les différents éléments du contexte de l’évaluation, dont: une politique de sécurité du système, une spécification des fonctions requises dédiées à la sécurité, une définition des mécanismes de sécurité (optionnelle), la cotation annoncée de la résistance minimum des mécanismes, et le niveau d’évaluation visé.

Les ITSEC proposent 10 exemples de classes de fonctionnalité prédéfinies [ITSEC 1991, §2.59-2.64, annexe A]:

- Les exemples de classes de fonctionnalité F-C1, F-C2, F-B1, F-B2, F-B3 sont les classes de confidentialité qui correspondent aux exigences de fonctionnalité des classes C1 à A1[1] dans les TCSEC.
- L’exemple de classe de fonctionnalité F-IN concerne les TOE pour lesquelles il y a des exigences élevées d’intégrité pour les données et les programmes, comme dans le cas des bases de données.
- L’exemple de classe de fonctionnalité F-AV impose des exigences élevées pour la disponibilité.
- L’exemple de classe de fonctionnalité F-DI impose des exigences élevées en ce qui concerne la préservation de l’intégrité des données au cours de leur transmission.
- L’exemple de classe de fonctionnalité F-DC est destiné aux TOE très exigeantes en matière de confidentialité des données au cours de leur transmission.
- L’exemple de classe de fonctionnalité F-DX est destiné aux réseaux très exigeants en matière de confidentialité et d’intégrité des informations.

Les différents critères d’assurance exigés se découpent en deux aspects: les **critères d’assurance d’efficacité**, et les **critères d’assurance de conformité**. Ces critères d’assurance se découpent ensuite à nouveau en deux catégories vis-à-vis de la construction et de l’exploitation du système. Les critères d’assurance de conformité sont définis vis-à-vis de 6 niveaux d’exigences, numérotés de E1 à E6, qui correspondent à des contraintes de plus en plus fortes et définissent le niveau de certification atteint par une TOE (pour la classe de fonctionnalité pour laquelle elle est évaluée). Le tableau 5 présente la définition des différents critères d’assurance d’efficacité pour les six aspects pris en compte dans les ITSEC (quatre vis-à-vis de la conception, deux vis-à-vis de l’exploitation).

Ces points sont particulièrement pertinents dans le cadre de ce mémoire car ils montrent que, dans le cadre des ITSEC, une cible d’évaluation peut être considérée comme sûre malgré l’identification de **vulnérabilités**, c’est-à-dire de faiblesses des mécanismes de sécurité, dans la TOE. Bien évidemment, ces faiblesses doivent pouvoir être jugées négligeables, au vu des conséquences possibles ou en raison de l’absence de menaces susceptibles de les exploiter, pour que le niveau visé soit atteint. Vis-à-vis de ce point, le manuel d’évaluation associé aux critères, ou ITSEM [ITSEM 1993], distingue deux types de mécanismes. Les mécanismes de type A sont ceux qui possèdent une certaine vulnérabilité dans leur principe même (comme

[1] Fonctionnellement, B3 et A1 sont identiques (cf. page précédente).

Catégorie	Aspects	Définition des critères d'assurance d'efficacité
Construction	Pertinence de la fonctionnalité	L'analyse de la pertinence doit montrer comment les menaces sont contrées par les fonctions et les mécanismes dédiés à la sécurité.
	Cohésion de la fonctionnalité	L'analyse de cohésion doit montrer qu'il est impossible d'amener l'une des fonctions ou l'un des mécanismes dédiés à la sécurité à rentrer en conflit ou à se mettre en contradiction avec d'autres fonctions ou mécanismes dédiés à la sécurité.
	Résistance des mécanismes	Même si un mécanisme dédié à la sécurité ne peut pas être court-circuité, désactivé, altéré ou contourné, il peut encore être possible de le mettre en échec par une attaque directe tirant profit des insuffisances de ses algorithmes, ses principes, ou ses propriétés sous-jacents. Pour cet aspect de l'efficacité, la capacité de ces mécanismes à contenir une telle attaque directe est estimée. Cet aspect de l'efficacité se distingue des autres en ce qu'il exige de prendre en considération le niveau des ressources qui seraient nécessaires à un agresseur pour réussir une attaque directe.
	Estimation de la vulnérabilité de la construction	Avant et pendant l'étude des autres aspects de l'évaluation, diverses vulnérabilités dans la construction (tels que des moyens de désactiver, court-circuiter, altérer ou contourner des fonctions et mécanismes dédiés à la sécurité) auront été identifiées. L'analyse de l'impact potentiel de chacune des vulnérabilités connues doit montrer qu'elle ne peut pas être exploitée parce que: (a) elle est convenablement couverte par d'autres mécanismes de sécurité non compromis, ou (b) il peut être montré que la vulnérabilité ne relève pas de la cible de sécurité, qu'elle n'existera pas dans la pratique, ou qu'elle pourra être convenablement contrée par des mesures de sécurité extérieures à la TOE (définies dans la documentation).
Exploitation	Facilité d'emploi	Cet aspect de l'efficacité examine si la TOE peut être configurée ou utilisée d'une manière qui n'est pas sûre, mais qu'un administrateur ou un utilisateur final pourrait raisonnablement croire sûre.
	Estimation de la vulnérabilité en exploitation	Avant et pendant l'étude des autres aspects de l'évaluation, diverses vulnérabilités dans l'exploitation de la TOE auront été identifiées. Comme précédemment, l'analyse de l'impact des vulnérabilités connues doit montrer qu'elles ne peuvent pas être exploitées.

Tableau 5 - Définitions des critères d'assurance d'efficacité des ITSEC

les algorithmes cryptographiques, ou les mécanismes d'authentification par mot de passe, basés sur un secret particulier). Les mécanismes de type B sont ceux qui n'auront aucune faiblesse, en tout cas s'ils sont parfaitement conçus et réalisés. Dans tous les cas, on peut estimer qu'une vulnérabilité n'est pas contrée par le système si on constate que des mécanismes ne sont pas *pertinents* et ne contrent pas

certaines menaces, ou ne sont pas *cohérents* et ne forment pas un ensemble intégré. Dans le cas d'une cible d'évaluation protégée par différents mécanismes, une vulnérabilité est représentée comme une réduction de la résistance des mécanismes à un des points de l'ensemble des protections qui entourent la cible d'évaluation. L'ITSEM ne considère pas vraiment le problème de vulnérabilités apparaissant du fait de la composition des faiblesses de *différents* mécanismes. En effet, la cotation de la résistance des mécanismes protégeant différents aspects de la cible d'évaluation est celle du mécanisme qui a la plus faible cotation. Dans le cas où des mécanismes combinés sont utilisés pour assurer une même protection (comme par exemple, dans le cas de l'utilisation simultanée d'une authentification par mot de passe et par empreintes digitales), c'est la résistance du mécanisme possédant la plus forte cotation qui est retenue. L'objectif est donc de faire en sorte que la résistance de chacun des mécanismes successifs entourant la cible soit supérieure à un minimum requis.

Trois niveaux de résistance à une attaque directe des mécanismes de sécurité sont définis dans les ITSEC. La résistance minimum de chaque mécanisme critique doit être cotée comme étant *élémentaire*, *moyenne* ou *élevée*. D'après les critères [ITSEC 1991, §3.5-3.8]:

- pour que cette résistance soit côtée *élémentaire*, il doit être manifeste que le mécanisme fournit une protection contre une subversion accidentelle aléatoire, bien qu'il soit susceptible d'être mis en échec par des agresseurs compétents;
- pour que cette résistance soit côtée *moyenne*, il doit être manifeste que le mécanisme fournit une protection contre des agresseurs dont les opportunités ou les ressources sont limitées;
- pour que cette résistance soit côtée *élevée*, il doit être manifeste que ce mécanisme ne pourra être mis en échec que par des agresseurs disposant d'un haut degré de compétence, d'opportunité et de ressources, une attaque réussie étant jugée irréalisable normalement.

L'ITSEM précise la manière dont on peut estimer le niveau de résistance des mécanismes de sécurité [ITSEM 1993, §3.3.29-32, §6.C.28-34]. Comme la résistance des mécanismes est relative à la *compétence*, aux *opportunités* et aux *ressources*, il est nécessaire de développer la signification de ces termes:

- La *compétence* représente la connaissance nécessaire aux personnes pour pouvoir attaquer une cible d'évaluation. Un *profane* est alors quelqu'un sans compétence particulière; une *personne compétente* est quelqu'un qui connaît les fonctionnements internes de la cible, et un *expert* est quelqu'un qui connaît bien les principes et algorithmes sous-jacents utilisés dans la cible.
- Les *ressources* représentent les ressources que l'attaquant doit employer pour attaquer avec succès sa cible. On s'intéresse principalement à deux types de ressources: le *temps* et l'*équipement*. Le temps est le temps passé par un attaquant pour réaliser une attaque, sans compter le temps d'étude. Les équipe-

ments comprennent des ordinateurs, des appareils électroniques, des outils matériels et du logiciel. Dans ce cadre :
  - le temps peut se répartir schématiquement entre *en quelques minutes*, *en quelques jours*, et *en quelques mois*.
  - *sans équipement* signifie qu'aucun équipement spécial n'est nécessaire ; un *équipement disponible* est un équipement disponible dans l'environnement d'exploitation de la cible d'évaluation, dans la cible d'évaluation ou dans le commerce ; un *équipement spécial* est un équipement spécifique pour réaliser une attaque.
- Les *opportunités* recouvrent des facteurs qui peuvent généralement être considérés comme hors du contrôle de l'attaquant, tels que les cas où l'assistance d'une autre personne est nécessaire (*collusion*), la possibilité d'un concours de circonstances particulier (*chance*) et la possibilité de l'interception de l'attaquant (*détection*). Ces facteurs sont difficiles à coter en règle générale. Dans les ITSEM, seules les formes suivantes de collusion sont considérées : *seul*, *avec un utilisateur*, et *avec un administrateur*. Cette définition de la collusion suppose que l'attaquant n'est pas un utilisateur autorisé de la cible d'évaluation.

Les ITSEM indiquent explicitement que les facteurs exposés ci-dessus ne sont pas supposés être définitifs, ni complets. Pourtant, ils constituent une tentative de fournir un moyen de qualifier individuellement le niveau de difficulté de mise en œuvre d'une vulnérabilité présente dans le système. Les ITSEM fournissent également des tables élémentaires permettant de déterminer le niveau de difficulté en effectuant un calcul quantitatif, même si les valeurs obtenues n'ont d'autre signification que de permettre de déterminer le niveau de la résistance d'un mécanisme (élémentaire, moyenne, ou élevée).

1.3.1.3 Les critères communs

Les critères communs [CC 1996a] sont nés de la tentative d'harmonisation des critères canadiens, des critères européens et des critères américains. La première version des *Common Criteria for Information Security Evaluation* a ainsi été largement distribuée, l'objectif étant de parvenir à la mise au point d'une deuxième version destinée à devenir une norme internationale.

Les critères communs contiennent deux parties bien séparées comme dans les ITSEC : fonctionnalité et assurance. De même que dans les ITSEC, les critères communs définissent également une cible d'évaluation (TOE). Une des différences essentielles entre ITSEC et critères communs réside dans l'existence des profils de protection (*Protection Profiles*) [CC 1996b] qui avaient auparavant été introduits dans les critères fédéraux américains [Federal Criteria 1992]. Un profil de protection définit un ensemble d'exigences de sécurité et d'objectifs, indépendants d'une quelconque implémentation, pour une catégorie de TOE. L'intérêt de ces profils est double : un développeur peut inclure dans la définition de la TOE un ou plusieurs profils de protection, un client désirant utiliser un système ou un produit peut également demander à ce que ce système corresponde à un profil de protection particu-

lier, évitant ainsi de donner une liste exhaustive des fonctionnalités et des assurances qu'il exige. Une partie importante des critères communs est donc consacrée à la présentation de profils de protection prédéfinis.

Cette notion de profil représente la volonté américaine qui consiste à préférer évaluer des systèmes qui entrent dans le cadre de profils connus. La tendance européenne serait plutôt de définir systématiquement une TOE et ses exigences de sécurité et d'évaluer cette TOE sans nécessairement s'appuyer sur des profils prédéfinis. Les critères communs tentent de réaliser un compromis équitable entre ces deux positions.

1.3.1.4 Des critères d'évaluation de la sûreté de fonctionnement

Il apparaît de plus en plus au sein de la communauté scientifique la volonté d'étudier de façon globale les problèmes de sûreté de fonctionnement, sans se limiter aux seuls attributs de la sécurité-confidentialité. Le projet SQUALE de la communauté européenne, dans le cadre du programme ACTS, est un exemple de cette volonté d'élaborer des critères d'évaluation pour la sûreté de fonctionnement dans son ensemble, en incluant les aspects relatifs à la sécurité-confidentialité, mais également à la sécurité-innocuité et aux autres attributs de la sûreté de fonctionnement.

La version préliminaire de ces critères [Corneillie *et al*. 1997] considère 4 niveaux de confiance que l'on peut attribuer à un système, notés de D1 à D4[1], où D1 est le plus faible niveau de confiance et D4 le plus fort. Un niveau D0 indique également l'absence d'exigences de sûreté de fonctionnement pour le système. À cette cotation globale sont associés des niveaux de confiance similaires pour chacun des attributs présentés au 1.1.1.1, comme indiqué dans le tableau 6.

Attribut	Niveaux
Disponibilité (*Availability*)	A1-A4
Confidentialité (*Confidentiality*)	C1-C4
Fiabilité (*Reliability*)	R1-R4
Intégrité (*Integrity*)	I1-I4
Sécurité-innocuité (*Safety*)	S1-S4
Maintenabilité (*Maintainability*)	M1-M4

Tableau 6 - Échelle des niveaux de confiance

Le projet SQUALE retient également une notion similaire à celle introduite dans les ITSEC et définit donc une **cible de sûreté de fonctionnement** contenant une cible d'évaluation et les différents éléments nécessaires à l'évaluation. Pour chaque système, on peut désirer atteindre un niveau de confiance différent pour chacun des attributs, constituant ainsi un **profil de sûreté de fonctionnement**, comme: A1, C0, R3, I3, S3, M2 (le niveau C0 indiquant ici que le système ne présente aucune exi-

1. La lettre D correspondant à "*Dependability*".

gence en termes de confidentialité). Le **niveau de confiance global** du système est défini comme le niveau de confiance le plus élevé devant être obtenu pour un des attributs. Dans le cas précédent, le niveau de confiance global du système est donc D3. C'est en fonction de ce niveau que le processus d'évaluation doit être effectué.

L'attribution d'un niveau de confiance à un système précis implique quatre domaines d'évaluation:

- l'évaluation des objectifs de sûreté de fonctionnement du système, afin de déterminer leur adéquation et leur pertinence;
- la vérification du système qui doit garantir la validité de l'implémentation des fonctions relatives à la sûreté de fonctionnement qui sont mises en œuvre dans le système;
- la validation de la sûreté de fonctionnement;
- et enfin l'évaluation de la qualité du processus de construction du système (qui doit permettre d'avoir confiance dans le développement et l'exploitation du système).

Chacun des critères d'évaluation appartenant à l'une de ces catégories peut être associé à un ou plusieurs niveaux de confiance, et devra être vérifié pour l'évaluation complète.

1.3.1.5 Conclusion

Dans l'annexe dévolue aux conseils pour les acheteurs de systèmes de sécurité, les ITSEM signalent [ITSEM 1993, §6.4.30] que "*Bien que ces lignes directrices puissent être suivies dans d'autres domaines, elles ont été développées pour des applications de type militaire et ne concernent principalement que l'aspect confidentialité de la sécurité. D'autres travaux sont nécessaires pour étendre la portée de ces lignes directrices à d'autres aspects de la sécurité et à d'autres domaines d'application*". En effet, malgré les éléments que les différents critères d'évaluation permettent de rassembler afin d'améliorer la confiance que l'utilisateur peut avoir dans le service rendu par le système (vis-à-vis de la sécurité), il n'en reste pas moins que le point de vue pris par l'ensemble des critères d'évaluation est fortement marqué par leur origine. Ceci explique d'une part l'accent mis sur la propriété de confidentialité, et d'autre part les inadéquations éventuelles avec les exigences de domaines d'application différents de leur domaine d'origine.

Nous noterons également que l'ensemble des critères d'évaluation ne donnent qu'une vision statique de la sécurité, même quand ils s'intéressent à la phase d'exploitation du système. Ils ne prennent donc pas en compte directement l'influence de l'utilisation de ces systèmes sur leur sécurité. Enfin, certains critères et notamment les ITSEC montrent que des vulnérabilités résiduelles peuvent persister dans le système même après son évaluation. Il est alors important de tenir compte de ces vulnérabilités et d'étudier leur impact sur la sécurité.

1.3.2 Analyse des risques

Dans les différents critères, l'évaluation de l'impact de vulnérabilités résiduelles sur la sécurité du système reste assez limitée. Notamment, aucune évaluation quantitative n'est proposée. L'accent est mis sur la prévention des fautes plutôt que sur la prévision. Malgré tout, les ITSEC estiment que ces aspects de prévision des fautes ne peuvent pas être écartés et imposent l'identification des vulnérabilités résiduelles et leur évaluation ordinale. Les méthodes d'*analyse des risques* se concentrent sur cette analyse et ont pour objectif d'évaluer plus précisément les conséquences sur la sécurité de l'existence de vulnérabilités résiduelles.

Dans la terminologie de l'analyse des risques, un *risque* peut être vu comme un événement redouté. On peut lui attribuer une fréquence d'occurrence et un coût. En ce sens, cette notion se rapproche de la notion de défaillance définie au 1.1.1. Dans ce domaine, on distingue également trois notions :

- l'*analyse des risques* qui est la discipline générale destinée à permettre la mise en œuvre d'une politique de sécurité en tenant compte des rapports coûts/bénéfices apparaissant dans un système ou une organisation ;
- l'*évaluation des risques* qui consiste à identifier les risques présents dans un système ou une organisation, et à évaluer les conséquences potentielles pour le système ou l'organisation et l'efficacité des parades associées ;
- la *gestion des risques* qui consiste à sélectionner les parades permettant de minimiser l'exposition des biens aux risques, compte tenu des compromis financiers, politiques ou sociaux nécessaires (dans le cadre d'une organisation, la gestion des risques est typiquement dévolue à des responsables possédant une vision globale de l'organisation).

L'analyse des risques est le concept communément considéré comme le plus large, et englobant les deux autres. L'ensemble des différentes méthodes regroupées sous cette dénomination doivent permettre de motiver un choix de moyens de protection à mettre en œuvre pour protéger les intérêts du système ou de l'organisation. Le lecteur trouvera dans [Dacier 1994] une analyse des différentes tendances existant dans le domaine, nous nous intéressons seulement ici aux éléments communs à l'ensemble de ces méthodes et caractéristiques de l'analyse des risques. Nous pouvons distinguer sept étapes dans cette approche :

- L'identification des actifs, qui représentent les éléments à protéger dans le système.
- L'identification des menaces : pour chaque actif identifié à l'étape précédente il est nécessaire d'identifier les menaces correspondantes.
- L'analyse des conséquences qui estime les conséquences de la réalisation d'une menace sur les actifs du système.
- Le calcul des risques : pour chaque menace on doit calculer le risque qu'elle représente, ce risque étant évalué comme égal à la fréquence d'occurrence de la menace multipliée par la conséquence financière de sa réalisation.

- L'évaluation des parades possibles qui identifie les contre-mesures utilisables pour parer à une menace ainsi que leur efficacité et leur coût.
- L'évaluation du niveau de sécurité existant qui, en fonction des résultats précédents, estime un niveau général de sécurité, et permet de proposer un choix de parades optimal du point du vue du rapport coût/efficacité pour améliorer ce niveau.
- La formulation d'un plan de sécurité qui constitue le résultat final de l'étude incluant un compte rendu des résultats précédents et un plan de mise en œuvre des nouvelles mesures de protection choisies.

La réalisation de ces différentes étapes peut s'effectuer sous divers modes opératoires suivant la méthode choisie. On peut distinguer les approches basées sur:

- les listes de contrôle qui partent des parades disponibles et étudient la nécessité de leur mise en place;
- l'évaluation quantitative (la méthode la plus classique) qui partant de l'identification exhaustive (si possible) des actifs, des vulnérabilités et des menaces, propose une quantification des risques;
- l'utilisation de questionnaires qui étudient les risques en se basant sur la vision des utilisateurs et des responsables du système (et dont l'objectif est d'arriver de ce fait à se concentrer sur les fonctions du système ou de l'organisation);
- l'identification de scénarios d'attaque, qui prennent en compte non seulement la possibilité de réussir avec succès à exploiter une vulnérabilité, mais également la probabilité de réussir à exploiter une succession d'attaques;
- l'utilisation de parades pré-définies, choisies à partir de l'expertise acquise sur le développement d'autres systèmes, et éventuellement réutilisées après une analyse de pertinence sur le système visé.

Le principal problème de l'analyse des risques reste celui de la collecte de données. Soit la méthode se veut exhaustive et devient alors extrêmement coûteuse en temps et en effort (sans pourtant offrir de réelle garantie méthodologique), soit elle utilise une méthode de sélection et s'en remet alors en pratique à la qualité de l'expertise de l'évaluateur. Ces problèmes de collecte de données nous semblent étroitement liés à l'absence d'une modélisation rigoureuse du système ou de l'organisation, indépendante de ceux-ci, en vue de l'étude des risques associés. De plus, l'absence d'une modélisation stricte ne permet pas de proposer des évaluations rigoureuses des conséquences des menaces et des vulnérabilités sur la sécurité du système. En effet, on ne tient pas compte des dépendances qui peuvent exister entre différents mécanismes de sécurité et entre différentes vulnérabilités pour évaluer les risques engendrés par une menace. Finalement, il découle de ceci que l'expertise et le savoir-faire de l'évaluateur sont des éléments indissociables de la méthode qu'il met en œuvre et que la qualité des résultats obtenus repose finalement sur lui.

1.3.3 Évaluation quantitative

À l'exception des méthodes d'analyse des risques, les approches dédiées à l'évaluation quantitative de la sécurité d'un système sont relativement peu développées. Toutefois, un certain nombre de travaux se sont penchés sur l'utilisation de méthodes d'évaluation rigoureuses permettant d'obtenir une quantification de la sécurité.

1.3.3.1 Mesure de l'information

Dans le cadre des systèmes basés sur une politique multi-niveaux, on a vu que le problème de l'identification et de l'élimination des *canaux cachés* constituait une part importante de la validation du système. Vis-à-vis de ce problème, un certain nombre d'approches ont été développées dans le but d'obtenir une évaluation quantitative des canaux cachés, en essayant, par exemple, d'estimer la bande passante de ces canaux [Karger & Wray 1991 ; Moskowitz 1992], ou encore de déterminer la quantité d'information susceptible de transiter par ce biais [Moskowitz & Miller 1994].

Dans le cadre de l'étude de la confidentialité des systèmes, les travaux présentés dans [Trouessin 1991b ; Trouessin 1991a] s'intéressent à l'évaluation quantitative de la préservation de la confidentialité sur la base d'un ensemble de mesures d'entropie. Ces mesures sont principalement utilisées pour comparer le comportement de différentes stratégies de *fragmentation* et de *traitement fragmenté* des données, par rapport à la préservation de la confidentialité.

Vis-à-vis de la fragmentation des données, cette technique permet d'effectuer le choix de la stratégie de fragmentation des données la plus adaptée vis-à-vis d'un ensemble d'opérations binaires (donc non fragmentées). Elle est appliquée dans [Trouessin 1991b] à des niveaux de granularité très fins correspondant à une fragmentation par tranches de bits.

Concernant le traitement fragmenté, cette approche a été appliquée à certains algorithmes issus du domaine du calcul parallèle et relatifs à la résolution de systèmes linéaires. Ceci a permis de mettre en évidence le niveau de fragmentation optimal pour chaque algorithme qui permet de minimiser les risques de non préservation de la confidentialité. Ces résultats montrent néanmoins la nécessité d'une analyse spécifique pour chaque implémentation fragmentée d'un algorithme.

Par ailleurs, bien que cette approche fournisse des résultats quantitatifs tout à fait pertinents vis-à-vis du problème abordé, elle se cantonne à la confidentialité, ainsi qu'à une technique tout à fait particulière d'obtention de propriétés de sécurité (et plus largement de sûreté de fonctionnement) pour un système informatique : la *fragmentation-redondance-dissémination* [Fabre *et al*. 1996]. Du point de vue de la confidentialité, cette technique impose certaines contraintes. Par exemple il faut interdire, sur chaque processeur impliqué dans le traitement global, toute vision

continue d'une partie de l'espace des données, afin d'empêcher toute induction d'information depuis un fragment du programme vers un ou plusieurs autres fragments du programme.

1.3.3.2 Mesures probabilistes

Des modèles probabilistes ont également été utilisés afin d'étudier les propriétés de sécurité de certaines catégories de systèmes [Lee 1989]. Ces travaux ont pris pour cible les systèmes correspondant aux différents niveaux de certification introduits par le livre orange [TCSEC 1985]. Leur principe est d'associer aux différents niveaux de classification des informations ou d'habilitation des individus des probabilités comme:

- $d_c(x)$ définie comme la probabilité qu'une information de niveau de classification c puisse causer des dommages de moins de x unités (dans une certaine échelle) si cette information est compromise;
- $h_{c_1, c_2}(x)$, définie comme la probabilité qu'un individu possédant un niveau d'habilitation c_2 et ayant accès à des informations de niveau de classification c_1, provoque *directement* moins de x unités de dommages en utilisant ces informations de manière malveillante;
- ou $r_{c_1, c_2}(t)$ définie comme la probabilité qu'un individu possédant un niveau d'habilitation c_1 et cible d'une menace de moins de t unités communique de l'information classifiée au niveau c_1 à une personne possédant seulement une habilitation de niveau c_2.

Dans le cas d'un système, seule la probabilité $r_{c_1, c_2}(t)$ est considérée, en utilisant des niveaux de *confiance* correspondant aux niveaux d'évaluation du livre orange (notamment, B2, B3 et A1).

Bien que les différents paramètres correspondant à ces probabilités soient restés, dans un premier temps, relativement arbitraires, l'utilisation du calcul des probabilités à partir de ces définitions permet d'étudier les risques associés à l'utilisation de tels systèmes. Un résultat notable montre que, sous certaines conditions, la combinaison de deux systèmes classés au niveau B2 des TCSEC et caractérisés par des modes de défaillances bien distincts, est aussi sûre qu'un système classé au niveau B3. Cette observation encourage à prendre en compte, non seulement la résistance d'un système individuel, mais également la résistance combinée de différents systèmes pour l'évaluation de la sécurité globale.

Un certain nombre de métriques ont également été proposées pour étudier l'influence de l'utilisation simultanée de plusieurs mécanismes ou de plusieurs sources dans le cadre de procédures complexes d'authentification pouvant contenir plusieurs chemins. Un chemin impliquant plusieurs entités intermédiaires dignes de confiance, où chacune est en mesure d'authentifier la suivante, est un cas de figure fréquemment rencontré dans les systèmes de grande taille. Plusieurs travaux se sont attachés à proposer des mesures permettant d'évaluer la confiance globale que l'on pouvait accorder à un tel processus d'authentification. La plupart de ces approches représentent le processus d'authentification sous la forme d'un graphe dont les

nœuds représentent les différentes entités apparaissant dans le processus, ou bien les différentes clefs cryptographiques impliquées dans le processus. Les arcs peuvent être de deux natures : soit une entité A considère qu'elle est en mesure d'authentifier une autre entité B (relation notée $A \rightarrow B$), soit une entité A fait confiance à une entité B pour authentifier d'autres entités ou pour lui recommander ces entités (relation notée $A \Rightarrow B$). Aux arcs du graphe sont associées des valeurs représentant la probabilité de succès ou la confiance accordée. À partir de ces valeurs, il est ainsi possible de proposer des mesures globales permettant d'évaluer la confiance relative existant entre deux nœuds et de quantifier le risque associé. On notera que dans le cas d'un système ouvert, cette confiance peut évoluer de façon non monotone quand de nouvelles relations de confiance apparaissent [Beth *et al*. 1994], ce qui renforce l'intérêt de l'utilisation d'une mesure permettant de la quantifier régulièrement. [Reiter & Stubblebine 1997] présente une vue d'ensemble des différentes méthodes existantes, ainsi qu'une mesure particulière intégrant une notion de coût.

1.3.3.3 Évaluation par utilisation du graphe des privilèges

L'approche présentée dans [Dacier 1994 ; Dacier *et al*. 1996] définit une méthode générale d'évaluation quantitative de la sécurité des systèmes informatiques. Elle est basée sur une représentation des vulnérabilités présentes dans un système informatique, appelée un **graphe des privilèges** [Dacier & Deswarte 1994]. Un **privilège** est défini comme étant un ensemble de droits Σ qu'un sujet s peut posséder sur un objet o. Les *nœuds* du graphe des privilèges représentent des ensembles de privilèges. L'existence d'un *arc* d'un premier ensemble de privilèges vers un second indique que la possession de ce premier ensemble permet d'acquérir le second, par application d'une ou plusieurs méthodes. En d'autres termes, l'existence d'un arc partant d'un nœud n vers un nœud n' signifie que tout sujet capable d'acquérir l'ensemble de privilèges représenté par n est aussi capable d'acquérir celui représenté par n', en utilisant la ou les méthodes définies par l'arc reliant n à n'. Ceci permet de tenir compte de la transitivité de la notion de cession (ou de vol) de privilèges. Les méthodes de transfert de privilèges à l'origine de l'existence des arcs dans le graphe correspondent à des **vulnérabilités** présentes dans le système. Ces vulnérabilités peuvent correspondre à des faiblesses du système (c'est-à-dire des fautes), mais peuvent également représenter des mécanismes de transfert de droits parfaitement licites et indispensables au fonctionnement du système. Chacun des arcs présents dans le graphe des privilèges est étiqueté par un identificateur correspondant à la méthode permettant d'effectuer le transfert de privilège entre les nœuds destination et origine de l'arc. La figure 4 présente un exemple simple de graphe des privilèges.

Dans le cas du système d'exploitation UNIX, les ensembles de privilèges correspondent aux droits d'accès des différents utilisateurs et des différents groupes d'utilisateurs définis dans le système. Par ailleurs, un certain nombre de méthodes licites ou illicites de cession de privilèges sont connues pour ce système d'exploitation [Dacier 1994 ; Dacier *et al*. 1996 ; Garfinkel & Spafford 1996]. La figure 5 présente

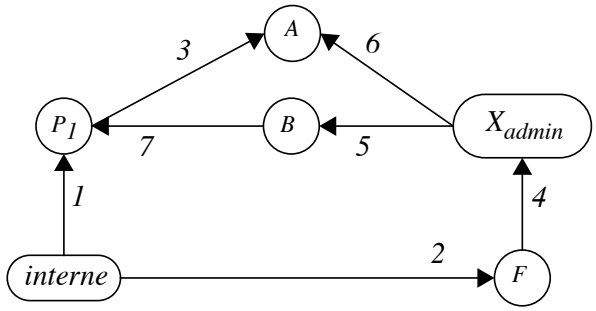
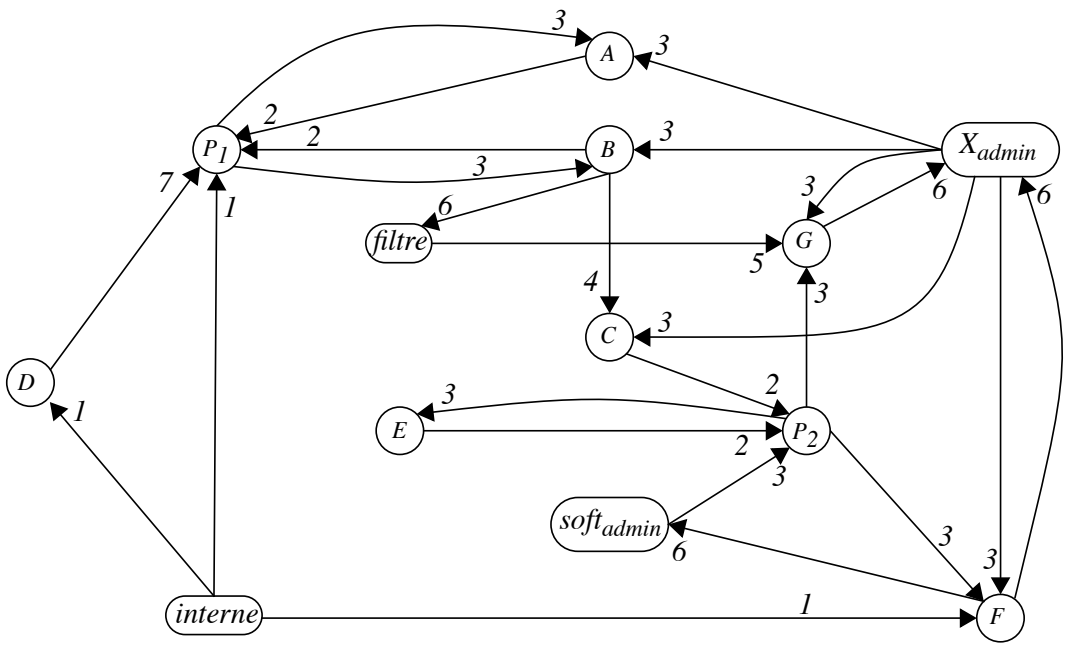


Figure 4 - Exemple de graphe des privilèges

un exemple théorique du type de graphe que l'on pourrait construire à partir de l'observation d'un tel système, en prenant en compte un certain nombre de ces vulnérabilités. En pratique, l'observation automatique et régulière d'un système UNIX par des outils logiciels spécifiques ou existants [Farmer & Spafford 1994] est parfaitement envisageable [Ortalo *et al*. 1997] et permet de construire automatiquement le graphe des privilèges correspondant.



1) "peut deviner le mot de passe de" ; 2) "est présent dans le .rhosts" ; 3) "peut modifier un répertoire présent dans la liste $PATH de" ; 4) "peut modifier le .xinitrc de" ; 5) "peut réaliser une attaque par le biais du courrier électronique" ; 6) "est un sur-ensemble des privilèges de" ; 7) "peut modifier l'exécution d'un programme dont le setuid bit est positionné qui appartient à".

Figure 5 - Exemple de graphe des privilèges appliqué au système UNIX

Étant donnée cette représentation, on peut envisager d'associer à chacune des vulnérabilités prises en compte lors de la construction du graphe des privilèges une *valeur numérique* correspondant à la probabilité de sa mise en œuvre. Dans

[Dacier 1994; Dacier *et al*. 1996], les différentes variables choisies pour évaluer ces probabilités correspondent au *temps* ou à l'*effort* nécessaires pour réussir à utiliser une vulnérabilité.

Dans cette vision, les objectifs de sécurité du système consistent à protéger un certain ensemble de privilèges, appelé la **cible**, d'un autre ensemble de privilèges, appelé l'**attaquant**. On peut donc envisager de définir une mesure quantitative de la sécurité comme étant la valeur de temps ou d'effort correspondant à la difficulté pour l'attaquant d'obtenir les privilèges de la cible, compte tenu de tous les chemins existants dans le graphe des privilèges, et des taux de transition associés à chaque vulnérabilité élémentaire qu'il devra éventuellement exploiter.

Une telle mesure est définie à partir de l'ensemble des différents scénarios qui peuvent être déduits du graphe des privilèges. Il est également nécessaire d'émettre un certain nombre d'hypothèses de haut niveau concernant le comportement de l'attaquant, détaillées dans [Dacier 1994; Dacier *et al*. 1996], qui conduisent à dire que, pendant la réalisation d'une intrusion:

- l'attaquant n'essaie pas d'exploiter des vulnérabilités qui l'amèneraient à obtenir un ensemble de privilèges qu'il possède déjà (hypothèse de bon sens);
- l'attaquant se souvient, tout au long de sa progression, des vulnérabilités qu'il n'a pas encore exploitées et peut donc revenir en arrière au cours de la réalisation de son intrusion (hypothèse de mémoire).

L'ensemble des scénarios d'intrusion que l'on peut identifier à partir d'un graphe des privilèges est appelé un **processus d'intrusion**. La figure 6 présente le processus d'intrusion que l'on peut construire à partir du graphe des privilèges de la figure 4 en considérant que l'attaquant est représenté par le nœud *interne* (noté *I* pour abréger), et la cible par le nœud *A* (en considérant les hypothèses de comportement précédentes), extrait de [Dacier 1994]. Dans cette figure, chaque nœud du processus d'intrusion correspond à une étape possible de l'attaque. Le processus d'intrusion forme un arbre dont les feuilles sont des nœuds contenant la cible (*A* dans notre exemple). Une branche de cet arbre correspond à un scénario d'attaque. Chaque nœud contient la liste des différents utilisateurs dont l'attaquant a obtenu les privilèges, et les différents arcs sont étiquetés par la méthode du graphe des privilèges permettant de passer d'un état à un autre.

La mesure quantitative de la sécurité utilisée dans [Dacier 1994; Dacier *et al*. 1996] est inspirée par des mesures classiques dans le domaine de la sûreté de fonctionnement, et plus précisément dans ce cas par le MTTF (Mean Time To Failure). Si on considère qu'un processus d'intrusion comme celui présenté dans la figure 6 peut être assimilé à un graphe de Markov (ici acyclique), on peut alors utiliser comme mesure quantitative de la sécurité la valeur du temps moyen nécessaire pour qu'un attaquant atteigne sa cible, définie par:

$$MTTF_k = T_k + \sum_{l \in out(k)} P_{kl} \times MTTF_l \quad \text{où} \quad T_k = \frac{1}{\sum_{l \in out(k)} \lambda_{kl}} \quad \text{et} \quad P_{kl} = \lambda_{kl} \times T_k$$

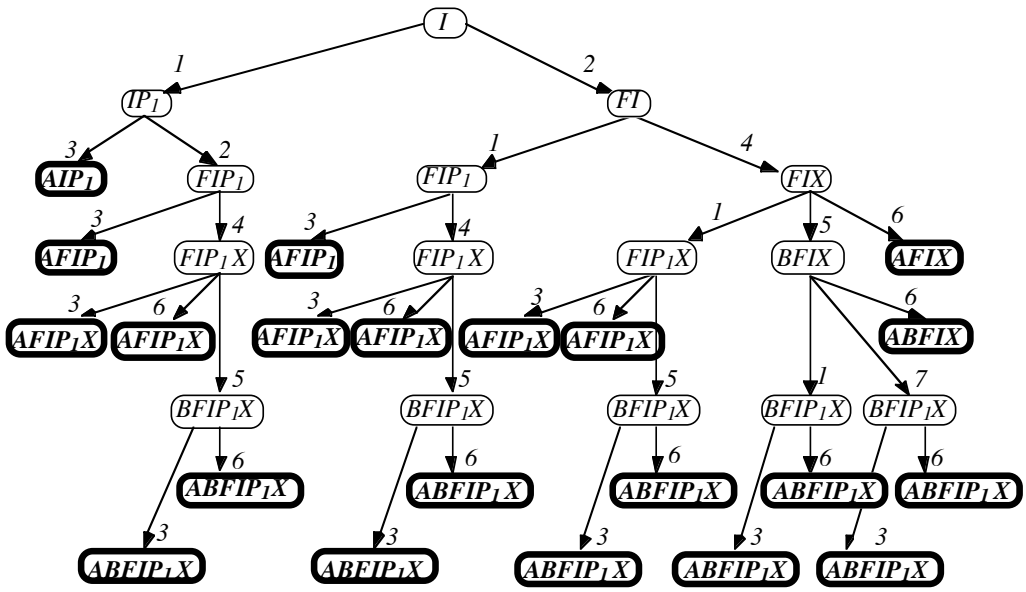


Figure 6 - Les différents états du processus d'intrusion

Dans cette équation, $MTTF_k$ représente le MTTF calculé en considérant l'état k comme l'état initial ; T_k représente le temps de séjour dans l'état k ; *out(k)* représente l'ensemble des états que l'on peut atteindre en une transition à partir de l'état k ; P_{kl} représente la probabilité de quitter l'état k pour l'état l ; λ_{kl} représente le taux de transition de l'état k vers l'état l (c'est-à-dire ici la valeur associée à chaque vulnérabilité élémentaire).

À partir du moment où on dispose d'une mesure quantitative de la sécurité du système représenté par son graphe des privilèges, on peut améliorer le résultat, dans le but d'améliorer la sécurité du système, en effectuant des modifications successives dans ce graphe et en observant l'évolution de la mesure. L'important ici est de disposer d'une justification précise et chiffrée des modifications que l'on propose pour améliorer la sécurité. [Dacier 1994] présente un exemple de ce processus visant à améliorer en plusieurs étapes les mesures de sécurité obtenues pour le système représenté par le graphe de la figure 5.

L'utilisation de cette méthode d'évaluation quantitative de la sécurité sur un système informatique de grande taille pendant plusieurs mois a permis de montrer qu'elle permettait effectivement d'identifier des événements ayant un impact significatif sur la sécurité du système [Ortalo *et al*. 1997]. La définition d'une mesure globale apporte donc réellement un moyen de contrôler la sécurité opérationnelle du système, en fournissant un indicateur reflétant son évolution, et en permettant d'identifier les événements à l'origine de ces évolutions. En revanche, la valeur absolue de cette mesure reste étroitement dépendante du système considéré, des vulnérabilités prises en compte, ainsi que des choix faits pour les valeurs associées à ces vulnérabilités (λ_{kl}). Ce sont essentiellement ses variations qui sont exploita-

bles. Dans le cadre de l'étude de la sécurité d'un système d'information, cette approche semble également utilisable. En effet, les notions de privilèges et de vulnérabilités s'étendent naturellement vers le cadre d'une organisation, ainsi que nous le verrons dans les chapitres suivants.

1.4 Conclusion

Les représentations des besoins de sécurité que nous avons passées en revue dans ce chapitre correspondent aux approches établies. Ainsi que nous avons pu le voir, la plupart des travaux possédant une base mathématique concernent la représentation de la sécurité d'un système informatique. En ce qui concerne la représentation des besoins de sécurité des organisations, les approches les plus utilisées restent généralement relativement informelles, à l'exception des organismes gouvernementaux qui sont en mesure de mettre en place des politiques de sécurité multi-niveaux. Il apparaît donc que les modèles formels utilisés pour étudier une politique de sécurité que nous avons présentés ne sont pas adaptés pour représenter les besoins de sécurité d'un système d'information au sens large, que ce soit par manque de souplesse ou par manque de rigueur. La tâche de définition des besoins de sécurité relatifs à un système d'information nécessite des développements supplémentaires. En particulier, il semble nécessaire de définir une méthodologie et un modèle formel de représentation nouveau afin de faciliter la définition de la politique de sécurité d'une organisation.

Cette politique de sécurité peut servir de base à la mise en œuvre d'une méthode d'évaluation de la sécurité. Ainsi que nous avons pu le voir, les principales méthodes d'évaluation couramment utilisées sont des méthodes d'évaluation ordinale basées sur des critères normalisés dans le cas des systèmes informatiques, ou sur des méthodes d'analyse des risques dans le cas des organisations. Toutefois, ces méthodes se concentrent sur l'évaluation statique du système. Dans chaque cas, le résultat de l'évaluation ne permet pas d'obtenir une évaluation de la sécurité opérationnelle du système, tenant compte des différents événements relatifs à la sécurité qui peuvent avoir lieu pendant son fonctionnement (comme une évolution de la configuration, ou les opérations effectuées par les utilisateurs). Pourtant, c'est généralement pendant cette vie opérationnelle que peuvent apparaître des vulnérabilités dangereuses pour la sécurité du système. La méthode d'évaluation de la sécurité basée sur le graphe des privilèges, présentée au 1.3.3.3, permet de tenir compte des différentes vulnérabilités existant dans le système comme de celles pouvant apparaître pendant son fonctionnement. De plus, elle permet d'obtenir une évaluation quantitative de la sécurité. L'évolution de ce résultat permet de détecter, parmi toutes les vulnérabilités ayant pu apparaître dans le système, celles qui ont un impact important sur la sécurité et qui nécessitent une action correctrice. Enfin, l'extension de cette méthode définie pour des systèmes informatiques en direction des systèmes

d'information semble possible, notamment si une définition rigoureuse de la politique de sécurité permet d'identifier les différents privilèges existant dans le système d'information.

Après avoir proposé une méthode de définition de la politique de sécurité appuyée sur le langage de la logique déontique qui semble suffisamment souple pour représenter les besoins de sécurité des différents systèmes d'information (chapitre 2), nous étudierons l'application à ces systèmes de la méthode d'évaluation de la sécurité basée sur le graphe des privilèges (chapitre 3).

Chapitre 2 Méthode de définition d'une politique de sécurité

Dans ce chapitre, nous proposons une méthode permettant de définir une politique de sécurité pour un système ou une organisation. Dans un premier temps, les étapes successives de la méthode permettant d'obtenir les différents éléments de la politique de sécurité sont présentées. Ensuite, nous proposons de représenter formellement cette politique de sécurité dans le langage de la logique déontique et nous montrons comment utiliser ce langage pour la spécifier. Nous donnons également diverses indications sur la manière dont cette spécification peut être exploitée. Enfin, nous nous intéressons à l'application de cette méthode, d'une part dans le cadre de la définition de la politique de sécurité d'une organisation, et d'autre part pour la représentation de certaines des politiques de sécurité utilisées dans les systèmes informatiques.

2.1 Structure de la méthode

Les différentes étapes de la méthode proposée afin de définir la politique de sécurité d'un système d'information (que ce soit celui d'une organisation ou qu'il s'agisse d'un système informatique) correspondent aux différents constituants de la politique de sécurité (cf 1.2.1, page 12). Dans une première étape, il est nécessaire d'obtenir certains éléments de description structurels ou fonctionnels indispensables pour la définition des règles ou des objectifs de sécurité. Ensuite, il s'agit d'ajouter les préoccupations de sécurité à cette description, c'est-à-dire préciser les règles et les objectifs de sécurité que l'on souhaite imposer dans le système d'information.

2.1.1 Description d'un système d'information

Certaines politiques de sécurité présentées précédemment, comme par exemple les politiques de contrôle d'interface (cf 1.2.7, page 27) font le choix de définir les propriétés de sécurité attendues du système indépendamment de la description du système lui-même. Le modèle utilisé pour représenter le système est alors un modèle très général (comme par exemple une machine à états) dans lequel on s'intéresse plus particulièrement aux évolutions affectant directement l'état de sécurité du système (par exemple, les transitions relatives à l'émission des sorties, ou celles affectant les niveaux de sécurité pris en compte dans le système). Cette approche permet

de faire largement abstraction des particularités du système d'information considéré dont la description n'est pas nécessaire [Lotz 1997]. De ce fait, la spécification des propriétés de sécurité attendues du système est indépendante de celui-ci. Néanmoins, la mise en œuvre d'une politique de sécurité de ce type conduit généralement à imposer des contraintes fortes sur le fonctionnement de l'organisation, ce qui soulève des difficultés.

Dans notre cas, contrairement à ces approches, la définition de la politique de sécurité intègre une description partielle du système considéré. Ceci permet d'utiliser un certain nombre des informations disponibles dans le système, et d'adapter la définition des objectifs ou des règles de sécurité à certaines contraintes imposées par la structure ou le fonctionnement du système (notamment dans le cas où celui-ci existe déjà). Les règles et les propriétés de sécurité peuvent donc, dans une certaine mesure, s'avérer plus détaillées et plus proches du fonctionnement réel.

Parmi les éléments de description du système que l'on peut vouloir intégrer à la politique de sécurité, on peut en distinguer deux types: les éléments de base qui permettent de désigner certaines entités spécifiques du système, et les éléments de description du fonctionnement qui permettent de prendre en compte certaines caractéristiques du fonctionnement du système.

2.1.1.1 Éléments de base

Les **éléments de base** de la politique de sécurité correspondent aux diverses entités du système concernées par la politique de sécurité. Par exemple, dans le cadre de la représentation d'une organisation, on voudra inclure dans la politique de sécurité la désignation des différents postes. De même, la politique de sécurité devra mettre en évidence: les différentes fonctions de l'organisation, les noms des différents individus appartenant à cette organisation, les différents matériels que l'on veut protéger (coffre-fort, documents papier, matières précieuses, appareils de production d'énergie, câbles de communication, etc.), et plus généralement la désignation de tout ce que contient l'organisation et qui peut être relatif à la sécurité. Dans le cadre d'un système informatique, les éléments de base appartenant en propre à la politique de sécurité désignent, par exemple: les différents programmes disponibles dans le système, les différents numéros d'identification associés aux comptes des utilisateurs, les noms des différentes machines, les groupes d'utilisateurs définis dans le système, etc. Ils peuvent également désigner des éléments qui ne sont pas explicitement décrits par des données du système, comme des rôles utilisés dans la mise en œuvre de la politique de sécurité, les tâches correspondant aux différents comptes d'un système informatique dans le cas où chaque compte n'est pas directement associé à un utilisateur (comme des comptes de développement ou de test), les actions élémentaires associées aux différentes étapes d'une opération complexe, etc.

Afin de représenter certaines politiques de sécurité, on peut également être amené à intégrer dans la politique de sécurité des éléments particuliers destinés à décrire l'état de sécurité du système. Ainsi, dans le cadre d'une politique multi-niveaux, on

devra inclure dans la politique la désignation de chacun des niveaux de sécurité pris en compte (et donc certainement des classifications du type CONFIDENTIEL, SECRET, TRÈS-SECRET, et des catégories du type NUCLÉAIRE, CHIMIQUE, BACTÉRIOLOGIQUE, IONIQUE) ou des éléments permettant d'effectuer la distinction entre un sujet, un objet, ou un droit (comme *lecture* et *écriture*).

On peut ébaucher une classification de ces différents éléments selon deux directions. Dans les exemples évoqués, on distingue en effet dans la spécification:

- les **éléments structurants** dont l'intérêt est de permettre de structurer la spécification (comme un groupe, un rôle, un bâtiment);
- et les **éléments atomiques** qui correspondent à des unités indivisibles (comme une classification, un droit, un individu).

Enfin, dans une politique de sécurité, on peut également identifier:

- les **éléments de sécurité**, qui désignent des éléments de description relatifs à l'état de sécurité du système (comme une classification, une catégorie, un droit);
- et les **éléments du système**, qui ne sont pas relatifs à son état de sécurité.

2.1.1.2 Règles de fonctionnement

Aux éléments de base qui constituent le vocabulaire sur lequel s'appuie la politique de sécurité viennent s'ajouter des règles de description du fonctionnement de l'organisation ou du système. Ces règles décrivent le fonctionnement général de l'organisation, tel qu'il doit être pris en compte dans la politique de sécurité.

Par exemple, dans un système informatique comme UNIX, l'accès aux données d'un fichier précis est conditionné par l'existence de certaines informations (appelées les droits d'accès) dans le système de fichiers. Un accès ne sera donc accordé à un utilisateur que si un certain nombre de conditions relatives à ces droits d'accès sont vérifiées. Toutefois, pour un utilisateur particulier appelé le super-utilisateur, des règles différentes, beaucoup moins contraignantes, s'appliquent. Une politique de sécurité s'appliquant à ce système d'exploitation devra donc contenir des règles de fonctionnement reflétant ce comportement (dans le cas simple d'un accès à un fichier d'une partition locale).

Dans le cas d'une organisation, ces règles de fonctionnement constituent une représentation des différents flux d'information. Les différentes tâches effectuées par les individus font évoluer les informations au travers de l'organisation (sans forcément impliquer des vérifications de l'état de sécurité). Cette évolution peut être représentée dans la politique de sécurité afin de tenir compte des contraintes liées au fonctionnement. Un exemple simple est celui d'une procédure d'achat auprès d'un organisme de distribution. Au sein de l'organisme, le processus suit plusieurs étapes, parmi lesquelles on distingue la réception de la commande et son enregistrement, puis l'accès au magasin et les opérations de facturation et de comptabilité (et, parallèlement, des opérations de gestion des stocks), suivis par l'expédition du bien (éventuellement conditionnée par la solvabilité du client). Ce processus peut varier

suivant que le client est un particulier ou un client nouveau qui paiera à la commande, ou une société déjà cliente de l'organisme pour laquelle la facturation sera généralement effectuée après l'expédition.

2.1.2 Description des objectifs de sécurité

Les objectifs de sécurité précisent les propriétés de sécurité (c'est-à-dire de confidentialité, d'intégrité et de disponibilité) qui sont attendues pour ce système. Il s'agit par exemple, de définir qu'une certaine catégorie d'information (qui peut être explicitement désignée, mais qui peut aussi être caractérisée par ses relations avec d'autres éléments) doit être inaccessible pour une certaine catégorie d'utilisateurs. L'ensemble de ces propriétés permet de déterminer quels sont les états du système qui sont considérés comme acceptables du point de vue de la sécurité. En poursuivant l'exemple du système informatique UNIX développé précédemment, un objectif de sécurité peut être d'interdire l'accès à un fichier particulier à tous les utilisateurs du système, à l'exception de quelques personnes explicitement désignées. Ces propriétés peuvent être complexes. Du point de vue de la confidentialité par exemple, un utilisateur peut avoir accès à une information sans pour autant avoir le droit de la diffuser à d'autres utilisateurs.

De manière générale, les objectifs de sécurité doivent déterminer précisément ce qui est permis, interdit et obligatoire dans le système. Un état de sécurité du système qui satisfait à l'ensemble des objectifs de la politique de sécurité du système est un état **sûr**.

2.1.3 Description des règles de sécurité

Dans certains cas, une politique de sécurité spécifie directement un certain nombre de règles qui doivent être mises en œuvre dans le système. Par exemple, le choix d'une politique multi-niveaux de Bell-LaPadula s'accompagne des deux règles de sécurité relatives aux opérations de lecture et d'écriture présentées au 1.2.5.1. Dans le cas général, la spécification de la sécurité, effectuée via la politique de sécurité, s'attache à préciser un certain nombre de mécanismes relatifs à la manière dont l'état de sécurité du système peut évoluer. De la même façon que dans la politique de Bell-LaPadula, les règles de sécurité permettent de contrôler la manière dont le système évolue du point de vue de la sécurité. Elles spécifient donc comment et dans quelles conditions l'état de sécurité du système peut être modifié. En général, ces règles sont relatives aux différents attributs de sécurité introduits comme éléments de base de la politique de sécurité (par exemple les compartiments). Dans l'exemple d'UNIX évoqué aux paragraphes précédents, une règle de fonctionnement correspond à la condition d'existence d'un droit d'accès dans le système de fichiers, imposée par les mécanismes du système d'exploitation pour la réalisation d'une opération. Une règle de sécurité liée à ce fonctionnement décrit la manière dont peuvent être positionnés ces droits. Sous UNIX, c'est le propriétaire du fichier qui peut mettre en œuvre les commandes permettant de modifier ces droits d'accès, et

donc de modifier l'état de sécurité du système. Le comportement de ces commandes reflète l'existence d'une règle de sécurité, caractéristique d'une politique de sécurité discrétionnaire. Dans le cas d'une organisation, les règles de sécurité définissent, par exemple, qui peut attribuer une nouvelle fonction à un individu dans l'organisation, ou sous quelles conditions un individu peut déléguer ses pouvoirs à d'autres individus.

2.1.4 Vérification de la cohérence

La politique de sécurité du système d'information, une fois définie, peut permettre d'effectuer un certain nombre de vérifications. On peut distinguer globalement quatre catégories d'incohérence [Ortalo 1998]:

- En effet, étant donné que les règles de sécurité du système permettent de savoir comment un état de sécurité peut évoluer, et que les objectifs de sécurité permettent de savoir si un état est sûr, il est a priori souhaitable que l'on puisse vérifier qu'il n'est pas possible, partant d'un état sûr et en respectant les règles de sécurité, d'atteindre un état non-sûr. Si c'est possible, la politique de sécurité n'est pas cohérente et devrait être modifiée. Mais cette vérification peut être difficile, car elle correspond à la résolution du problème de protection, indécidable dans le cas général [Harrison *et al*. 1976] (cf 1.2.4.1.1, page 17).
- De la même manière, une politique de sécurité contenant plusieurs objectifs de sécurité différents, il peut s'avérer que ces objectifs soient contradictoires.
- Par ailleurs, plusieurs règles de sécurité présentes dans la politique peuvent elles-même se contredire. Par exemple, un individu peut se trouver confronté à une interdiction et à une obligation vis-à-vis de la même action (par exemple du fait d'un cumul de rôles).
  Ces deux derniers cas signifient que la politique de sécurité est incohérente et impossible à mettre en œuvre, et donc, soit que les objectifs de sécurité doivent être revus, soit qu'il est nécessaire d'adjoindre de nouvelles règles permettant de lever l'incohérence (par exemple, en introduisant des priorités [Brown 1994]).
- Enfin, il est possible que les règles de fonctionnement entrent en conflit avec les objectifs et les règles de sécurité qui ont été définis. Dans ce cas, on identifie certaines des faiblesses de l'organisation, en montrant que, compte tenu du fonctionnement de cette organisation, il est possible de contourner les règles de sécurité. Ainsi que nous l'avons déjà mentionné, face à ce type de problème deux attitudes peuvent être envisagées: on peut choisir de modifier le fonctionnement de l'organisation (et donc décider d'adapter l'organisation aux mécanismes de sécurité utilisés), ou bien de déterminer de nouveaux mécanismes de sécurité compatibles avec le fonctionnement de l'organisation (et, dans ce cas, on adapte la politique de sécurité au système d'information considéré).

Du seul point de vue de la spécification, certains de ces cas de figure ne sont pas admissibles, notamment parmi les deux premières catégories. Néanmoins, un certain nombre des politiques de sécurité définies dans les organisations présentent de

telles caractéristiques [Cuppens & Saurel 1996]. Notamment, la situation où les règles de sécurité ne permettent pas de prouver que les objectifs du système sont vérifiés peut être malgré tout considérée comme une situation admissible dans le cas où les états non-sûrs identifiés sont extrêmement improbables ou très difficiles à atteindre. Toutefois, des informations complémentaires sont alors nécessaires pour qualifier plus complètement ces états avant d'accorder confiance au système. Le cas où les règles de sécurité du système sont incompatibles est plus ennuyeux du point de vue de la spécification, car il rend difficile l'exploitation de la politique de sécurité. Si ces incohérences ne révèlent pas une faute dans la définition de la politique de sécurité, l'adjonction de nouvelles règles permettant de lever l'incohérence globale devient nécessaire dans la politique de sécurité et viendra compléter la spécification. Les cas particuliers ainsi identifiés caractérisent alors certains des éléments du système d'information qui sont particulièrement délicats à traiter du point de vue de la sécurité.

2.2 Spécification

La définition d'une politique de sécurité est facilitée par l'approche structurée que nous avons présentée qui permet d'identifier les différents éléments la constituant. Toutefois, la mise en évidence précise des propriétés de sécurité attendues d'une organisation ou d'un système d'information suppose un effort supplémentaire de mise en forme. Par ailleurs, pour refléter précisément les besoins de sécurité du système d'information et pour servir de support éventuel à une vérification ou une évaluation de ce système vis-à-vis de la sécurité, elle doit rester rigoureuse. Compte tenu de ces besoins, nous envisageons à présent une méthode de spécification appuyée sur le langage formel de la logique déontique qui correspond à cet objectif.

Après avoir précisé l'intérêt d'utiliser une approche formelle, nous présentons dans cette section le langage logique que nous proposons d'utiliser. Nous précisons comment ce langage s'applique à la spécification d'une politique de sécurité, et nous illustrons son application pratique à la formalisation de la politique de sécurité d'une organisation, puis d'un système informatique.

2.2.1 Intérêt d'une approche formelle

S'il est souvent possible d'identifier des incohérences dans le texte des règlements de sécurité utilisés dans les organisations, c'est généralement en raison des ambiguïtés inhérentes à l'utilisation du langage naturel. Malgré toute l'attention qui peut être portée à la rédaction de ces documents, leur application peut être sujette à interprétation. Par ailleurs, la structure de la politique de sécurité telle qu'elle a été présentée au 2.1 n'est pas toujours utilisée pour rédiger ce document qui ne fait pas toujours la distinction entre la définition des objectifs de sécurité du système et l'identification de règles de sécurité à observer afin d'obtenir les propriétés attendues. Le principal atout de l'utilisation d'une approche formelle dans la spécifica-

tion d'une politique de sécurité réside dans l'élimination d'un certain nombre des ambiguïtés de la spécification. En effet, une spécification formelle est associée à un langage et à une sémantique bien précis qui éliminent la majeure partie des ambiguïtés résultant de l'utilisation du langage naturel. En revanche, l'utilisation d'un langage formel ne préserve pas, a priori, de l'introduction de fautes dans la spécification, et ne facilite pas particulièrement la rédaction si ce n'est par l'effort de structuration qu'il impose.

Un autre atout d'un langage formel réside dans la possibilité de manipuler la spécification en suivant des règles mathématiques, éventuellement avec l'assistance d'un outil de preuve. La possibilité d'effectuer des démonstrations ou de vérifier la validité de certaines propriétés en se basant sur un langage logique est bien évidemment une opportunité digne d'intérêt. On peut notamment envisager d'effectuer à partir de la spécification formelle de la politique de sécurité, les différentes vérifications de cohérence citées précédemment (cf 2.1.4) et tenter, d'une part de vérifier que les objectifs de sécurité ne sont pas incompatibles, et d'autre part de démontrer que les règles de sécurité ne permettent pas de mettre en défaut ces objectifs. Toutefois, on devra garder à l'esprit que, par exemple, ce dernier point correspond dans certains cas à la résolution du problème de protection. Or, dans le cas général ce problème est indécidable. On comprend donc que les démonstrations et les manipulations formelles que l'on peut être en mesure d'effectuer à partir de la spécification de la politique de sécurité peuvent être limitées par de tels résultats, ou tout simplement par l'état actuel des connaissances en matière de démonstration et l'état de l'art des démonstrateurs automatiques [Rushby 1993].

Le choix d'un langage formel pour la spécification d'une politique de sécurité s'effectue tout d'abord en fonction du domaine d'application bien précis de ce langage. Il semble indispensable de choisir un langage permettant de représenter naturellement des notions comme celles d'obligation et d'interdiction, que l'on retrouve dans une politique de sécurité. En même temps, ce langage doit permettre de représenter simultanément des objectifs de sécurité, c'est-à-dire des propriétés attendues pour le système, et un état de sécurité réel, qui peut éventuellement ne pas satisfaire ces propriétés. Ceci correspond au cas où l'on souhaite exprimer dans la spécification les actions devant être effectuées en cas de violation de la sécurité (préface de [Meyer & Wieringa 1993], [Santos & Carmo 1993]. Ces différentes remarques sont à l'origine de notre intérêt pour un langage logique particulier qui est utilisable malgré ces exigences assez particulières: la logique modale, et plus particulièrement une de ses branches, la logique déontique [Glasgow *et al*. 1990 ; Bieber & Cuppens 1992 ; Jones & Sergot 1992 ; Cuppens 1993a ; Cholvy & Cuppens 1997].

2.2.2 Utilisation d'une logique modale

La logique modale est la logique de la *nécessité* et de la *possibilité*. Plus précisément, elle constitue un cadre formel pour l'étude de ces notions, en fournissant, outre une représentation explicite de celles-ci par des opérateurs modaux (□ pour la

nécessité, $\Diamond$ pour la possibilité), la possibilité d'étudier leurs aspects intensionnels et déductifs dans le cadre de la logique formelle, ainsi que leurs aspects extensionnels, par la sémantique des mondes possibles (ou sémantique de Kripke). On peut attribuer la paternité de la logique modale à Aristote dans l'*Organon* (322 av.J.C.) [Aristote 1992, §I.3, §I.13, §I.8-I.22]. La notion de modalité doit pourtant être prise dans un sens plus général: on peut en effet caractériser les propositions modales par la présence de modes, c'est-à-dire d'éléments modificateurs sur les énoncés. La logique modale s'étend alors à de nombreuses modalités, parmi lesquelles on peut trouver:

- les modalités *ontiques*:
  "*Il est (nécessaire, possible, contingent, impossible) que p*"
- les modalités *temporelles*:
  "*Il (sera, a été) (toujours, à un moment donné) vrai que p*"
- les modalités *épistémiques*:
  "*x (sait, croit, doute) que p*"
- les modalités *dynamiques*:
  "*Il est (nécessaire, possible, impossible) en faisant … que p*"
- et les modalités *déontiques* qui nous intéressent plus particulièrement:
  "*Il est (obligatoire, permis, interdit) que p*"

La plupart des modalités indiquées ci-dessus ont donné lieu à la définition de diverses logiques modales, c'est-à-dire à différentes théories modales obtenues suivant le type de modalité considéré: logique ontique, temporelle, dynamique, épistémique, déontique. On réserve alors le terme de logique modale pour désigner la théorie mathématique de ces diverses théories modales. Celles-ci partagent en effet un certain nombre de caractéristiques communes.

La définition de la notion de nécessité pose des difficultés pour la définition d'un modèle sémantique. Une des innovations de la logique modale [Kripke 1959; Kripke 1963] s'appuie sur l'introduction d'un modèle sémantique comprenant un ensemble de mondes différents dans lesquels on considère la valeur de vérité des différentes propositions et notamment des propositions contenant un opérateur modal grâce à la définition que:

> "*Nécessairement p*" est vraie dans un monde *w* si et seulement si *p* est vraie dans tous les mondes w' directement accessibles depuis *w*.

L'idée ici est que tous les mondes ne sont pas forcément directement (ou même indirectement) accessibles depuis un monde donné *w*. Un monde *w* permet d'accéder directement à un monde w' seulement si toutes les propositions qui sont vraies dans w' sont possibles dans *w*. De la même manière, si une proposition est nécessairement vraie dans un monde *w*, elle doit être vraie dans tous les mondes w' auxquels *w* permet d'accéder. L'idée d'utiliser une relation d'accessibilité sur un ensemble de mondes a ouvert l'étude de la logique modale. On établit alors des liens entre les propriétés de cette relation d'accessibilité et le fait que certaines propositions modales soient (logiquement) vraies.

La conception traditionnelle de la logique modale est d'abord basée sur la définition d'un langage contenant certaines variables propositionnelles $a, b, c, \ldots$ constituant les phrases atomiques. Des phrases complexes sont ensuite définies et peuvent prendre les formes suivantes : $\neg p$ ("p n'est pas vraie"), $p \Rightarrow q$ ("si p, alors q" ou "p seulement si q"), et $\Box p$ ("nécessairement p"), où p et q sont n'importe quelles phrases (pas nécessairement atomiques). Les autres connecteurs logiques sont définis à partir de ces opérateurs élémentaires, à l'instar de la logique propositionnelle classique (en exceptant $\Box p$).

Enfin, on peut définir la notion de *possibilité* (opérateur $\Diamond$) à partir de celle de nécessité par : $\Diamond p = \neg \Box \neg p$. On dit que l'opérateur $\Diamond$ est un opérateur **dual** de $\Box$.

On donne également un nom aux deux dernières combinaisons de l'opérateur modal avec la négation, sans aller jusqu'à les noter spécifiquement : $\Box\neg$ est l'*impossibilité*, et $\neg\Box$ est l'*éventualité*.

L'étape suivante est de définir un modèle (ou une interprétation) pour ce langage. Typiquement, un modèle M est défini comme un triplet $\langle \mathrm{W}, \mathrm{R}, \mathrm{V} \rangle$, où W est un ensemble non vide de mondes, R une relation d'accessibilité entre ces mondes, et V une fonction qui associe à chaque phrase atomique p et à chaque monde w une valeur de vérité dans $\{\text{vrai, faux}\}$. C'est à partir de ce modèle que l'on peut définir et étudier la logique modale sous ses aspects sémantiques.

On trouvera dans l'annexe A (page 153) une présentation plus complète et plus rigoureuse de la logique modale et d'un certain nombre de résultats associés. Parmi les références classiques dans ce domaine, on peut mentionner [Hughes & Cresswell 1968 ; Chellas 1980], ainsi que [Catach 1989] pour une étude particulièrement intéressante en direction des langages *multimodaux*. Dans le cadre de notre étude sur la spécification d'une politique de sécurité, nous nous intéressons plus particulièrement à la logique déontique, dont un certain nombre des domaines d'application potentiels sont présentés dans [Meyer & Wieringa 1993]. La logique déontique introduit trois opérateurs modaux notés **O**, **P** et **F**, présentés ci-après, et respectivement lus comme l'*obligation*, la *permission* et l'*interdiction*.

2.2.3 Définition du langage

2.2.3.1 Langage de la logique déontique

La **logique déontique**, tout comme la logique modale, est construite comme une extension de la logique propositionnelle usuelle à laquelle un ou plusieurs opérateurs déontiques sont ajoutés. Plus précisément, si Φ est un ensemble de **propositions atomiques** $a, b, c, \ldots$, si $\neg$, $\vee$, $\wedge$, $\Rightarrow$ et $\Leftrightarrow$ désignent les **connecteurs booléens** habituels, et si **O**, **P** et **F** désignent les trois **opérateurs modaux** de la logique déontique, le langage de la logique déontique, noté $L_{\mathbf{O}}(\Phi)$ ou plus simplement $L_{\mathbf{O}}$ quand il n'y a pas d'ambiguïté, est l'ensemble des *formules* (ou *expressions*) construit par les règles suivantes:

- si $p \in \Phi$, p est une formule,
- si p et q sont des formules, $\neg p$, $p \vee q$, $p \wedge q$, $p \Rightarrow q$, et $p \Leftrightarrow q$ sont des formules,
- si p est une formule, $\mathbf{O}p$, $\mathbf{P}p$ et $\mathbf{F}p$ sont des formules.

De manière équivalente, mais plus synthétique, $L_{\mathbf{O}}$ est donc le langage généré par la règle de grammaire suivante, donnée en notation EBNF ("Extended Backus Normal Form"), où f désigne une formule de $L_{\mathbf{O}}$ et a un élément de Φ :

$$f ::= a \mid \neg f \mid f \vee f \mid f \wedge f \mid f \Rightarrow f \mid f \Leftrightarrow f \mid \mathbf{O}f \mid \mathbf{P}f \mid \mathbf{F}f \qquad (1)$$

Nous appelons une **formule modale** une formule de $L_{\mathbf{O}}$ contenant au moins un des opérateurs modaux **O**, **P** et **F**. Une formule non-modale est une formule n'en contenant aucun.

La traduction en langage naturel des formules $\mathbf{O}p$, $\mathbf{P}p$ et $\mathbf{F}p$ est, respectivement, "*Il est obligatoire que p*", "*Il est permis que p*" et "*Il est interdit que p*". Les opérateurs **O**, **P** et **F** représentent donc les notions d'**obligation**, de **permission** et d'**interdiction**. À l'instar des définitions des opérateurs de nécessité (opérateur $\Box$), de possibilité (opérateur $\Diamond$) et d'impossibilité dans la logique modale classique (cf 2.2.2) on peut énoncer des liens entre ces différents opérateurs[1]. Tout d'abord, on pourrait considérer que l'opérateur **F** représentant l'interdiction peut être défini à partir de l'opérateur **O** d'obligation, car l'obligation et l'interdiction sont liées par la relation suivante:

$$\mathbf{F}p = \mathbf{O}\neg p \qquad (2)$$

Cette relation signifie, en langage naturel, que l'interdiction de p est équivalente à l'obligation de "non p".

1. Tout comme il existe des liens entre les connecteurs booléens habituels qui permettent d'en choisir seulement deux (traditionnellement $\neg$ et $\vee$ ou $\neg$ et $\Rightarrow$) comme connecteurs fondamentaux, puisque:

$$\begin{cases} p \wedge q = \neg(\neg p \vee \neg q) \\ p \Rightarrow q = \neg p \vee q \\ p \Leftrightarrow q = \neg(\neg p \vee \neg q) \vee \neg(p \vee q) \end{cases} \qquad \begin{cases} p \vee q = \neg p \Rightarrow q \\ p \wedge q = \neg(p \Rightarrow \neg q) \\ p \Leftrightarrow q = \neg((p \Rightarrow q) \Rightarrow \neg(q \Rightarrow p)) \end{cases}$$

L'opérateur **P** est le *dual* de l'opérateur **O**. On peut également considérer la relation (3) entre les opérateurs **O** et **P**, qui relie les notions d'obligation et de permission de la même manière que sont reliés les opérateurs de nécessité et de possibilité de la logique modale classique.

$$\mathbf{P}p \;=\; \neg\mathbf{O}\neg p \tag{3}$$

Nous terminons la définition du système logique utilisé par la suite en précisant l'axiomatisation et les règles d'inférence[1] correspondant au langage $\mathrm{L}_{\mathbf{O}}$ choisi. Un système de logique modale est avant tout une extension de la logique propositionnelle, il contient tous les axiomes et toutes les règles d'inférences de la logique propositionnelle. Il contient donc l'ensemble des tautologies et il est fermé pour la règle d'inférence du *Modus Ponens*:

$$\frac{p \Rightarrow q,\ p}{q} \qquad \textit{Modus Ponens}$$

Nous considérons ici un système de logique modale **normal**, c'est-à-dire qui contient:

- l'axiome K:

$$\mathbf{O}(p \Rightarrow q) \Rightarrow (\mathbf{O}p \Rightarrow \mathbf{O}q) \qquad \mathrm{K}$$

- et la règle d'inférence RN (Règle de Nécessité[2]):

$$\frac{p}{\mathbf{O}p} \qquad \mathrm{RN}$$

La **logique déontique standard** (la plus communément utilisée), contient également l'axiome D qui relie les notions d'obligation et de permission. Cet axiome correspond à la volonté d'inclure dans la logique le fait qu'une obligation doive impliquer la permission correspondante.

$$\mathbf{O}p \Rightarrow \mathbf{P}p \qquad \mathrm{D}$$

La sémantique associée à une logique modale normale comme celle que nous venons de définir pour le langage logique $\mathrm{L}_{\mathbf{O}}$ est appelée **sémantique de Kripke**, ou encore **sémantique des mondes possibles** [Kripke 1963]. Un modèle de Kripke M pour un système de logique modale normal est un triplet $\langle \mathrm{W}, \mathrm{R}, \mathrm{V} \rangle$ où W est un ensemble de mondes possibles *w*, R est une relation binaire sur W appelée la **relation d'accessibilité**, et $\mathrm{V} : \mathrm{W} \times \Phi \rightarrow \{\text{vrai, faux}\}$ est une fonction qui donne, pour

1. Un **axiome** est un élément qui est défini comme faisant partie du langage. Une **règle d'inférence** possède la forme générale $\frac{f_1, \ldots, f_n}{f}$ dans lequel $f_1, \ldots, f_n$ désignent les *hypothèses* (ou *prémisses*) de la règle, et f sa *conclusion*. Un ensemble de formules est *fermé* pour une règle d'inférence — ou tout simplement *contient* cette règle — dans le cas où cet ensemble contient la conclusion de la règle chaque fois qu'il en contient les hypothèses.
2. "*Rule of Necessitation*"

chaque monde $w \in \mathrm{W}$ la valeur de vérité $\mathrm{V}(w, p)$ de la proposition atomique p. On note $\models^{\mathrm{M}}_{w} p$ le fait que la proposition p soit vraie dans un monde w dans le modèle M. Cette valeur de vérité est définie de la manière suivante, qui étend le calcul propositionnel habituel[1] :

- si $p \in \Phi$, $\models^{\mathrm{M}}_{w} p$ si et seulement si $\mathrm{V}(w, p) = \text{vrai}$,
- $\models^{\mathrm{M}}_{w} \neg p$ si et seulement si on n'a pas $\models^{\mathrm{M}}_{w} p$,
- $\models^{\mathrm{M}}_{w} (p \vee q)$ si et seulement si $\models^{\mathrm{M}}_{w} p$ ou $\models^{\mathrm{M}}_{w} q$,
- $\models^{\mathrm{M}}_{w} \mathbf{O}p$ si et seulement si $\forall w' \in \mathrm{W} / w\mathrm{R}w'$, $\models^{\mathrm{M}}_{w'} p$.

De cette définition, il découle donc que la formule "Il est obligatoire que p" est vraie dans un monde w si et seulement si "p" est vraie dans tous les mondes w' avec lesquels w est en relation. Quand w est en relation avec w' , on dit que w' est **directement accessible** depuis w, ou encore que w' est un **monde possible** pour w. La définition de la valeur de vérité de $\mathbf{P}p$ se déduit facilement de celle de $\mathbf{O}p$ grâce à (3):

- $\models^{\mathrm{M}}_{w} \mathbf{P}p$ si et seulement si $\exists w' \in \mathrm{W} / w\mathrm{R}w'$, $\models^{\mathrm{M}}_{w'} p$.

Dans le cas de la logique déontique standard incluant l'axiome D, on peut également remarquer que la relation R entre les mondes du modèle de Kripke est une relation *sérielle*, c'est-à-dire qu'on a: $\forall x \in \mathrm{W}, \exists y \in \mathrm{W} / x\mathrm{R}y$.

On peut définir des axiomatisations équivalentes pour une logique modale normale [Chellas 1980, §4.1], et notamment celle incluant la définition (3) et la règle d'inférence RK.

$$\frac{f_1 \wedge \ldots \wedge f_n \Rightarrow f}{\mathbf{O}f_1 \wedge \ldots \wedge \mathbf{O}f_n \Rightarrow \mathbf{O}f} \qquad \text{RK}$$

Dans la littérature, l'égalité (3) et la règle RK ne sont pas forcément incluses dans une définition de la logique déontique car leur prise en compte peut poser des problèmes d'interprétation sémantique qui peuvent justifier le rejet d'un système modal *normal*, notamment dans les cas où l'on souhaite représenter clairement la notion d'obligation conditionnelle [Chellas 1980, §6.5]. Toutefois, dans notre cas, nous nous intéressons avant tout à l'utilisation du langage déontique pour la spécification des politiques de sécurité. L'étude de la logique déontique en tant que telle, et en tant que langage de représentation des notions de droit (ou de sécurité), bien que pouvant avoir un impact direct sur nos travaux, sort du cadre de ce mémoire. Nous prendrons donc comme base initiale de notre langage une définition identique à celle du système KD de la logique modale classique, et notamment, nous acceptons (3) comme une définition acceptable de la notion de permission. Toutefois, cette simplification est avant tout faite dans le but de clarifier la méthode de spécifi-

1. En prenant comme connecteurs booléens élémentaires $\neg$ et $\vee$ (cf note 1, page 66).

cation d'une politique de sécurité d'une part, et dans l'optique de permettre par la suite un enrichissement de la structure du langage logique utilisé pour prendre en compte les enseignements des études consacrées à l'application de la logique déontique pour la formalisation de la sécurité, comme [Jones & Sergot 1992 ; Cuppens 1993b; Jones & Sergot 1993; Cuppens 1994a; Cuppens 1994b].

2.2.3.2 Extension du langage

Le langage $\mathrm{L}_{\mathbf{O}}$, ou $\mathrm{L}_{\mathbf{O}}(\Phi)$, est défini à partir de l'ensemble de propositions atomiques Φ. La définition de l'ensemble des propositions atomiques et des formules déontiques qui permettent de représenter une politique de sécurité dans ce langage peut être malcommode en raison de la nécessité de détailler toutes les propositions atomiques nécessaires pour construire la spécification ainsi que toutes les formules associées. Dans le but de faciliter la tâche de spécification, il est souhaitable d'introduire des éléments permettant une description structurée et hiérarchique des différentes formules de $\mathrm{L}_{\mathbf{O}}$. Dans cette optique, nous proposons d'utiliser une extension syntaxique, notée $\mathcal{L}_{\mathbf{O}}$, du langage précédent [Ortalo 1998 ; Ortalo & Deswarte 1998a].

Soit Π un ensemble d'ensembles $A, B, C, \ldots$, ordonné par la relation d'ordre partiel $\subseteq$ d'inclusion. Le langage $\mathcal{L}_{\mathbf{O}}(\Pi)$, ou plus simplement $\mathcal{L}_{\mathbf{O}}$ quand il n'y a pas d'ambiguïté, est le langage $\mathrm{L}'_{\mathbf{O}}(\Pi)$ défini par (1).

Cette extension reste purement syntaxique, et son intérêt réside dans la possibilité de structurer les propositions atomiques du langage. Par ailleurs, afin de faciliter également la construction des différents ensembles E de Π, on autorise l'utilisation de l'opérateur $\times$ (produit cardinal) afin de construire les ensembles de Π, avec évidemment $E \times E' = \{(e, e') / e \in E, e' \in E'\}$, et l'extension naturelle de la relation d'inclusion $E \times E' \subseteq F \times F' \Leftrightarrow E \subseteq F \wedge E' \subseteq F'$.

Le langage déontique $\mathrm{L}_{\mathbf{O}}(\Phi_{\Pi})$ correspondant au langage étendu $\mathcal{L}_{\mathbf{O}}(\Pi)$ est obtenu en considérant l'ensemble Φ_{Π} défini par (4). Φ_{Π} est donc l'ensemble des différents ensembles *terminaux* de Π, qui ne contiennent pas d'autres ensembles et constituent donc la base de la hiérarchie définie par la relation $\subseteq$.

$$\Phi_{\Pi} = \{E \in \Pi / (\forall E' \in \Pi, E' \not\subset E)\} \tag{4}$$

À une formule f de $\mathcal{L}_{\mathbf{O}}(\Pi)$ correspond un *ensemble* Σ_f de formules de $\mathrm{L}_{\mathbf{O}}(\Phi_{\Pi})$ défini récursivement par (5), où $f|_{x \leftarrow y}$ désigne la formule obtenue en substituant dans la formule f toutes les occurrences du symbole x par le symbole y. Une formule f de $\mathcal{L}_{\mathbf{O}}(\Pi)$ correspond donc en fait à une notation synthétique de plusieurs formules de $\mathrm{L}_{\mathbf{O}}(\Phi_{\Pi})$ qui sont toutes celles que l'on peut obtenir en remplaçant dans f toutes les occurrences d'ensembles non-terminaux de Π par leur contenu, jusqu'à obtenir des formules contenant uniquement des éléments de Φ_{Π}.

$$\Sigma_f = \bigcup_{E' \subseteq E} \Sigma_{f|_{E \leftarrow E'}} \tag{5}$$

Cette notation utilisant des ensembles correspond à des techniques déjà utilisées dans le domaine des langages logiques appliqués aux bases de données [Beeri *et al.* 1987 ; Kuper 1987 ; Kuper 1988 ; Shmueli *et al.* 1988 ; Beeri *et al.* 1991], ou dans le contexte d'applications appuyées sur un langage logique [Joubert *et al.* 1982]. L'objectif de cette extension est de faciliter la tâche de l'administrateur de sécurité chargé d'élaborer la spécification, en lui permettant d'abord de définir progressivement et hiérarchiquement les différentes propositions atomiques du langage grâce aux différents ensembles, et ensuite d'utiliser directement ces ensembles pour rédiger les formules logiques décrivant la politique de sécurité. Afin d'éclairer l'utilisation pratique de cette notation, nous présentons un certain nombre de formules écrites dans le langage étendu $\mathcal{L}_{\mathbf{O}}(\Pi)$ et l'ensemble de formules équivalent dans $\mathrm{L}_{\mathbf{O}}(\Phi_{\Pi})$. Dans ces exemples, les ensembles *A*, *B*, 'Administrateurs', et 'Services' désignent des ensembles contenant uniquement des propositions atomiques, 'Installation' et 'Logiciel' désignent directement des propositions atomiques.

À une équation simple du langage étendu $\mathcal{L}_{\mathbf{O}}(\Pi)$, par exemple $\mathbf{P}(A \vee B)$, correspond donc tout un ensemble de formules dans le langage $\mathrm{L}_{\mathbf{O}}(\Phi_{\Pi})$, ici l'ensemble $\{\forall p \in A, \forall q \in B, \mathbf{P}(p \vee q)\}$.

L'utilisation du produit scalaire pour construire des ensembles structurés dans $\mathcal{L}_{\mathbf{O}}(\Pi)$ permet de construire facilement des éléments, par exemple à partir de certaines de leurs caractéristiques. Ainsi, la formule (6) énonce directement tout un ensemble de permissions relatives à des actions d'installation de logiciels, caractérisées par le symbole de l'action 'Installation' (proposition atomique, c'est-à-dire définie par un ensemble vide), les individus susceptibles de l'accomplir qui sont ici regroupés dans l'ensemble des administrateurs du système, et certains logiciels objets de ce type d'action. Ces derniers sont également décrits à partir de deux caractéristiques : leurs types (traitement de texte, tableurs, etc.) regroupés dans l'ensemble 'Types de logiciels', et les services de l'organisation qui les détiennent.

$$\mathbf{P}(\text{Installation} \times \text{Administrateurs} \times (\text{Types de logiciel} \times \text{Services})) \qquad (6)$$

La description de tout ces éléments dans le langage d'origine $\mathrm{L}_{\mathbf{O}}(\Phi_{\Pi})$ nécessiterait la définition d'un grand nombre d'éléments, partageant pourtant des caractéristiques communes. L'équation (7) indique l'ensemble des formules de $\mathrm{L}_{\mathbf{O}}(\Phi_{\Pi})$ qu'il faudrait définir une à une afin de décrire les mêmes permissions. On voit que la contrepartie du produit scalaire dans une notation conventionnelle correspondrait à l'utilisation de plusieurs indices.

$$\{\forall a \in \text{Administrateurs}, \forall s \in \text{Services}, \forall t \in \text{Type de logiciel}, \quad \mathbf{P}(\text{Installation}_{a,\, \text{Logiciel}_{t,\, s}})\} \qquad (7)$$

Afin d'améliorer la souplesse de la politique de sécurité, il est possible d'introduire des éléments comme les rôles ou les groupes dans la spécification. La formule (6) présentée précédemment montre qu'il est possible dans $\mathcal{L}_{\mathbf{O}}$ d'utiliser directement des ensembles afin de définir et d'utiliser des groupes d'utilisateurs. Pour introduire des rôles, il faut énumérer à la fois les différents rôles, les différents utilisateurs, et

les différentes permissions existantes, mais également les paires (utilisateur, rôle) désirées. On peut alors modifier les règles de sécurité afin de définir les permissions accordées au utilisateurs en fonction des rôles qui leur ont été attribués. Ainsi, on peut modifier (6) en (8).

$$\mathbf{P}(\text{Installation} \times \text{Agent} \times (\text{Types de logiciel} \times \text{Services})) \Rightarrow \text{Agent} \times \text{Administrateur système} \quad (8)$$

Dans cette solution, les rôles apparaissent comme des éléments intermédiaires entre les permissions existants dans le système et les utilisateurs. Ces entités supplémentaires améliorent la souplesse de la spécification, car les rôles peuvent être définis indépendamment des éléments qu'ils relient. Cette flexibilité est obtenue au prix d'un effort de spécification supplémentaire. Ainsi qu'on peut le voir dans [Ortalo 1997], plusieurs variantes de cette méthode peuvent être utilisées pour introduire des rôles dans une spécification en logique déontique.

L'utilisation de cette notation peut poser certaines difficultés dans le cas où le même ensemble apparaît plusieurs fois dans une même formule logique. Ainsi, on peut envisager de faire correspondre à la formule $A \wedge B \vee C \wedge B$ de $\mathcal{L}_{\mathbf{O}}(\Pi)$ deux ensembles différents de formules de $\mathrm{L}_{\mathbf{O}}(\Phi_{\Pi})$:

1) soit $\{\forall p \in A, \forall q \in B, \forall r \in C, p \wedge q \vee r \wedge q\}$;
2) soit $\{\forall p \in A, \forall q \in B, \forall r \in C, \forall s \in B, p \wedge q \vee r \wedge s\}$.

Dans notre cas, la traduction opérée par (5) correspond au cas 1. Donc, chaque fois qu'un même ensemble apparaît dans une formule, il désigne le même élément. Si le besoin de définir des formules correspondant au cas 2 survient, il est alors nécessaire de définir un nouvel ensemble B' égal à B et d'utiliser la formule $A \wedge B \vee C \wedge B'$ à la place de la formule originelle. Dans cette situation, l'ensemble B' peut-être considéré comme une copie de B.

2.2.3.3 Utilisation d'une représentation graphique

Afin de faciliter la conception et la manipulation d'un ensemble de formules écrites dans le langage $\mathcal{L}_{\mathbf{O}}$, il est possible d'utiliser une représentation graphique dont l'intérêt pratique peut s'avérer significatif si cette représentation graphique est supportée par un outil d'édition. Outre le fait qu'une présentation agréable et une manipulation intuitive des formules logiques constitue un atout ergonomique, la définition et la manipulation de l'ensemble Π structuré par la relation $\subseteq$ se prête particulièrement à une représentation graphique sous forme d'une hiérarchie de listes. Nous présentons donc rapidement ci-après les conventions utilisées pour représenter les différents éléments du langage sous forme graphique. On notera que, dans cette représentation, pour des commodités d'édition, il est également possible de regrouper plusieurs formules au sein d'un ensemble nommé.

La figure 7 montre un exemple de représentation des propositions atomiques du langage, en s'appuyant sur la description hiérarchisée de $\mathcal{L}_{\mathbf{O}}$. La partie gauche de la figure présente un exemple de définition d'ensembles (éléments de Π). L'ensemble et ses différents éléments sont présentés sous la forme d'une liste hiérarchique indi-

quant les différentes relations d'inclusion. La partie droite de la figure illustre la définition d'un autre ensemble, construit à partir du produit cardinal de deux des ensembles précédemment introduits.

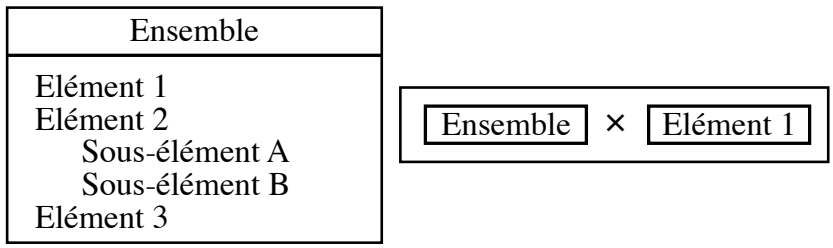


Figure 7 - Représentation hiérarchique des propositions atomiques

La figure 8 est un exemple de la définition de deux opérateurs logiques : l'opérateur déontique **O** et l'opérateur ⇒. Il s'agit là simplement d'une réécriture des formules textuelles usuelles.

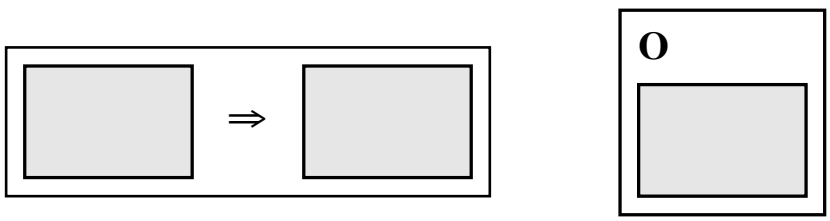


Figure 8 - Représentation des opérateurs logiques et déontiques

Enfin, la figure 9 présente un exemple d'une formule simple écrite en utilisant les éléments précédents.

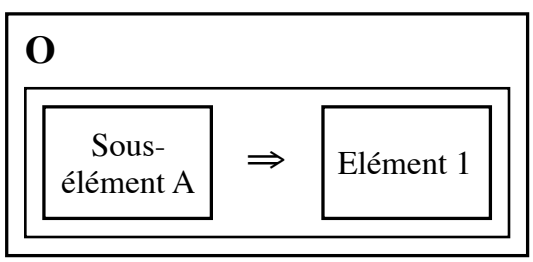


Figure 9 - Exemple de représentation d'une formule

L'avantage de la représentation graphique réside dans les possibilités de manipulation offertes par un outil d'édition qui permet de manipuler les définitions d'ensembles, et de désigner les arguments des opérateurs par un simple glisser-déposer. Des motivations analogues ont déjà été mentionnées dans [Heydon *et al*. 1990]. Un outil utilisant cette approche a été développé afin de montrer son intérêt pour la spécification de politiques de sécurité en logique déontique [Laffont & Ortalo 1997]. Cet outil est basé sur la librairie graphique Amulet [Myers *et al*. 1997a ; Myers *et al*. 1997b], dont les concepts fondamentaux (un modèle objet ins-

tance-prototype et l'utilisation de contraintes liant les objets graphiques) s'adaptent particulièrement bien aux représentations utilisées. La figure 10 présente un écran de travail de cet outil, contenant la définition de plusieurs ensembles et de quelques formules modales correspondant à des objectifs de sécurité. Les éléments graphi-

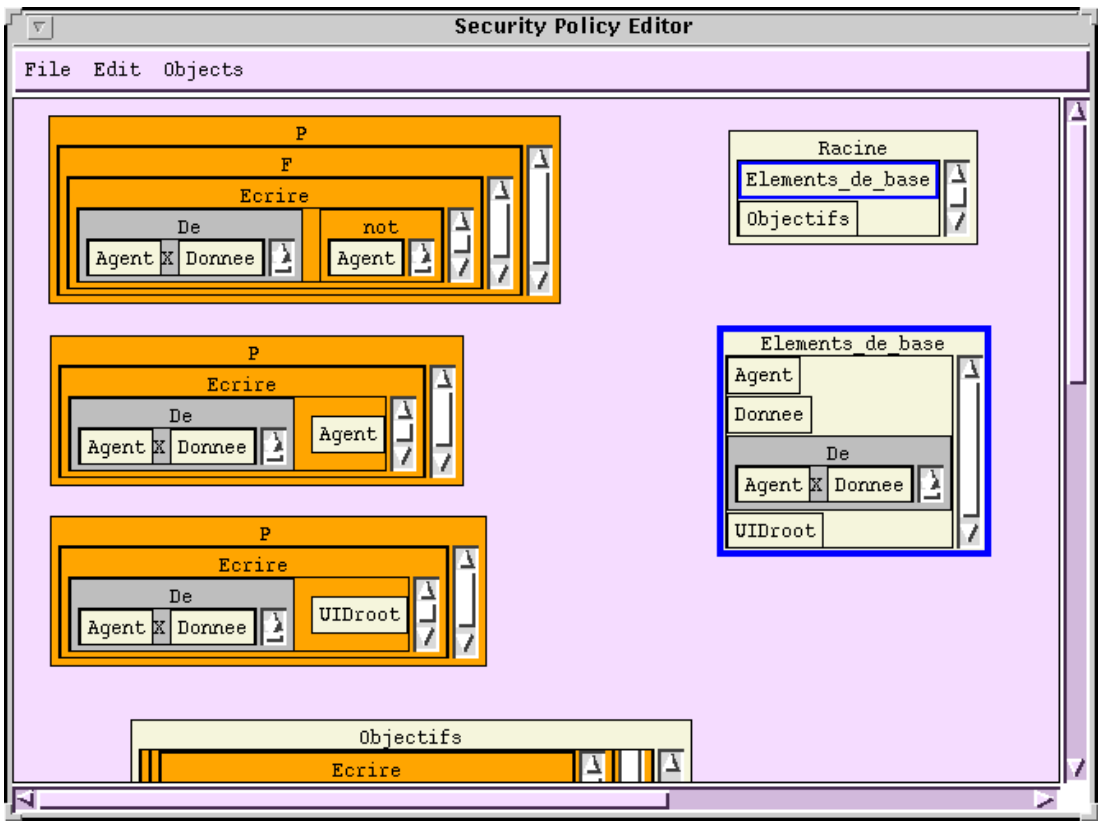


Figure 10 - Outil graphique de spécification d'une politique de sécurité

ques représentant des ensembles simples (à droite dans la fenêtre) sont constitués par le nom de l'ensemble (par exemple `Elements_de_base`) encadrant la liste des différents éléments directement contenus dans cet ensemble. Le produit scalaire de plusieurs ensembles est représenté en notation infixe, et peut aussi être nommé (par exemple `De` qui représente les données de tous les différents agents). Un opérateur est un objet graphique présentant un nombre fixe d'emplacements (associés aux différents arguments) dans lequel peuvent être placés des références à des ensembles, ou à d'autres opérateurs. Initialement, des prototypes d'opérateurs doivent être définis par l'utilisateur en utilisant un objet spécifique. Les prototypes des différents opérateurs sont disponibles pour l'édition dans une liste particulière, et deviennent immuables dès leur première utilisation.

Chacun des objets graphiques peut être sélectionné avec la souris, et déplacé vers un autre élément de la spécification. Une nouvelle référence à l'objet d'origine est alors insérée dans l'élément cible de la manipulation, ce qui revient à indiquer

l'argument d'un opérateur, ou à ajouter le contenu d'un ensemble à l'intérieur d'un autre. Il est également possible de déplacer un ensemble d'un élément vers un autre en pressant certaines touches pendant l'opération. Un certain nombre de contrôles sont effectués par l'outil pendant l'édition pour éliminer certaines constructions qui conduisent à des situations aberrantes. Ainsi, il n'est pas possible d'ajouter le contenu d'un ensemble à un de ses sous-ensembles stricts, ou d'utiliser un opérateur dont tous les arguments ne sont pas définis comme argument d'un autre opérateur, etc. Un certain nombre d'améliorations pourraient être apportées à cet outil au niveau de la spécification des différents opérateurs du langage logique. Notamment, il semble désirable d'introduire, au niveau des prototypes d'opérateurs, un domaine de définition (contrôlé par l'éditeur) pour chacun des arguments afin d'identifier certaines erreurs de manipulation assez courantes.

2.2.4 Formalisation d'une politique de sécurité

Le langage ainsi défini est bien adapté à la formalisation d'une politique de sécurité. Nous détaillons ci-après, pour chacun des éléments d'une politique de sécurité telle que celle que nous avons présentée au 2.1, les éléments du langage qui permettent de les représenter. Nous soulignons également les différentes propriétés formelles que l'on peut souhaiter obtenir dans le cas où la spécification est utilisée comme support pour la vérification de la politique de sécurité.

2.2.4.1 Éléments de description

2.2.4.1.1 Éléments de base

Dans le langage de spécification, les éléments de base de la politique de sécurité sont représentés par les propositions atomiques. Par exemple, à chaque individu membre de l'organisation ou à chaque machine appartenant au système informatique, on associe un certain nombre de propositions atomiques pour les désigner. Chacun de ces cas permet également d'illustrer l'intérêt d'utiliser des éléments structurants dans la description, comme c'est possible dans le langage étendu de spécification. En effet, plutôt que d'énumérer directement l'ensemble des individus d'une organisation, ou l'ensemble des machines composant un réseau informatique, il est préférable de définir différents groupes correspondant à différentes parties de l'organisation ou du système informatique. Ceci facilite l'élaboration et la compréhension de la spécification. Cela facilite également l'utilisation des éléments de base pour la définition ultérieure des propriétés de sécurité, qui concernent généralement une partie de l'organisation. Enfin, on est également amené à définir dans les éléments de base de la description des éléments abstraits qui peuvent représenter, par exemple, les différentes actions[1] effectuées par les individus dans l'organisation, ou les différentes tâches qu'une machine réalise (comme par exemple, les opérations *lecture* et *écriture* définies dans le modèle de Bell-LaPadula).

2.2.4.1.2 Règles de fonctionnement

La description des règles de fonctionnement du système nécessaires pour la définition de la politique de sécurité s'effectue, dans le langage $\mathcal{L}_{\mathbf{O}}$, par le biais des différents opérateurs de la logique propositionnelle traditionnelle. Il s'agit donc essentiellement de définir une représentation déclarative des différentes relations causales pouvant exister dans l'organisation ou le système considéré. Par exemple, on pourra vouloir intégrer le fait qu'un individu est matériellement en mesure de réaliser une action seulement si certaines conditions sont vérifiées dans le système, ou qu'une machine ne peut effectuer une opération que si les données d'entrées lui sont fournies. De même, si dans l'organisation différentes actions, ou dans un système informatique certaines transformations, s'enchaînent automatiquement, il faudra éventuellement intégrer à la spécification la chaîne causale correspondante. Ceci est notamment nécessaire si certaines étapes intermédiaires du processus font apparaître des éléments figurant dans les objectifs de sécurité.

Cette représentation du fonctionnement reste assez limitée. Il faut garder à l'esprit le fait qu'elle n'est intégrée à la spécification de la politique de sécurité que dans les cas où elle est pertinente vis-à-vis des objectifs de sécurité. Une représentation complète et détaillée du fonctionnement du système n'est pas nécessaire car elle ne serait pas utilisée pour étudier les règles et les objectifs de sécurité. En revanche, cette représentation doit être cohérente: si des règles de fonctionnement sont contradictoires l'ensemble de la spécification devient inutilisable car les objectifs de sécurité s'appliquent alors à un système vide (au sens où le système logique représenté ne peut être que celui où tous les éléments de base sont faux).

Du point de vue sémantique, les règles de fonctionnement définissent la structure interne des différents mondes du modèle de Kripke M associé à la spécification. Ce sont en effet des axiomes ne contenant pas d'opérateurs modaux (c'est-à-dire des formules non-modales), qui n'ont donc aucun impact sur les caractéristiques de la relation R entre les mondes du modèle. En revanche, chaque monde est tel qu'il constitue un état de fait compatible avec les différentes règles de fonctionnement. Ainsi, si on a défini une règle de fonctionnement du type $p \wedge q \Rightarrow r$, on sait que dans tous les mondes w où p et q seront vraies, r le sera aussi.

1. On notera que la notion d'*action* pourrait également être une notion modale. L'utilisation d'une logique déontique de l'action est d'ailleurs largement étudiée dans la littérature [Lindahl 1977; Pörn 1977; Hilpinen 1993; Cuppens 1994b]. Toutefois, malgré la manière dont elle est définie classiquement qui permet de regrouper les opérateurs modaux déontiques et dynamiques pour les traiter ensemble, en toute rigueur ceci nous amènerait à discuter de l'utilisation d'une logique multimodale [Catach 1989]. Malgré son intérêt, sur lequel nous revenons dans l'annexe A (page 153), nous excluons délibérément ce cas dans notre approche.

2.2.4.2 Objectifs de sécurité

2.2.4.2.1 Description

La description des objectifs de sécurité du système est associée à l'utilisation des opérateurs modaux **O**, **P** et **F** dans la spécification. Les opérateurs modaux permettent de modifier les propriétés de la relation R entre les différents mondes du modèle M associé à la spécification. Ils indiquent ainsi si deux mondes (caractérisés par des états de fait différents) doivent ou non être accessibles l'un depuis l'autre. Une formule du type $\mathbf{F}p$ indique ainsi qu'aucun des mondes accessibles dans le modèle (directement ou indirectement) ne doit permettre de conclure que p est vraie dans ce monde. Elle correspond donc du point de vue de la sécurité au souci d'interdire une propriété. La formule $\mathbf{P}p$ au contraire, signifie qu'à partir de n'importe lequel des mondes, on doit pouvoir atteindre un monde dans lequel p est vraie, et correspond donc à une permission. L'intérêt de ces opérateurs réside également dans la possibilité d'utiliser des formules complexes comme argument des opérateurs. Ainsi, la formule $\mathbf{F}(p \wedge q)$ caractérise un souci d'exclusion entre deux états (comme on peut le trouver dans la politique de la muraille de Chine, cf 1.2.8.2, page 34). La formule $\mathbf{O}(p \Rightarrow q)$ indique qu'une règle de fonctionnement du système est considérée comme obligatoire. Il est également possible de combiner deux opérateurs modaux et, par exemple, la formule $\mathbf{PF}p$ représente la notion complexe consistant à *permettre d'interdire* une situation. Ceci signifie qu'à partir d'un monde w, il doit être possible d'atteindre un deuxième monde w' depuis lequel il n'est en revanche pas possible d'atteindre un troisième monde w'' dans lequel p soit vraie.

Les objectifs de sécurité du système représentent les propriétés attendues par le système du point de vue de la sécurité. De la même manière, les formules modales représentent des contraintes imposées à la relation existant entre les mondes du modèle M. Ces formules conduisent donc à considérer une famille de relations R respectant ces contraintes (comme les relations réflexives, transitives, etc.), et donc une classe des modèles, valides vis-à-vis de ces formules modales. Ce sont donc ces formules que nous utiliserons pour représenter les objectifs de sécurité.

2.2.4.2.2 Propriétés nécessaires

Comme nous l'avons vu précédemment (cf 2.1.4) on peut vouloir effectuer un certain nombre de vérifications concernant la cohérence des objectifs définis dans la politique de sécurité. Ces vérifications correspondent à la vérification de la validité de certaines formules vis-à-vis de la politique de sécurité. Par exemple, on peut souhaiter prouver que les formules (9) et (10) ne sont *pas* vraies dans la politique de sécurité spécifiée.

$$\mathbf{O}p \wedge \mathbf{F}p \tag{9}$$

$$\mathbf{P}p \wedge \mathbf{F}p \tag{10}$$

D'ores et déjà, on peut dire que dans le système logique que nous avons considéré, la formule (10) est fausse de par les définitions de **P** et **F** données par (3) et (2) car $\mathbf{P}p \wedge \mathbf{F}p \Leftrightarrow \neg\mathbf{O}\neg p \wedge \mathbf{O}\neg p$ et $\neg\mathbf{O}\neg p \wedge \mathbf{O}\neg p$ est faux. On peut également remarquer que (9) est fausse dans le cas où on considère un système de logique déontique incluant l'axiome D[1]. Les contraintes de cohérence que l'on impose sur la politique de sécurité correspondent donc à un choix d'axiomes fondamentaux pour le système de logique modale utilisé. Les propriétés de plusieurs des différents systèmes modaux étudiés dans la littérature sont présentées dans [Hughes & Cresswell 1968; Chellas 1980], mais le choix définitif du meilleur système modal pour la spécification d'une politique de sécurité est un problème ouvert. Il est d'ailleurs tout à fait possible que, comme dans la majorité des cas d'utilisation de la logique modale, il n'existe pas d'axiomatisation idéale mais que celle-ci dépende en fait des besoins [Alchourròn 1993]. Par exemple, l'intégration de l'axiome D dans la spécification semble poser problème au vu des nombreuses incohérences que l'on observe en pratique dans les règlements actuels des organisations. Il serait souhaitable que les incohérences correspondant à D soient levées, mais cela peut éventuellement être impossible pour des raisons internes à l'organisation et dans ce cas, la politique de sécurité devrait pouvoir les tolérer et se dispenser de cet axiome [Chisholm 1963].

2.2.4.3 Règles de sécurité

Les liens entre les situations existant dans chacun des mondes *w* du modèle M (dont les différentes règles de fonctionnement régissent la structure), et les propriétés structurelles générales désirées pour ce modèle (qui sont décrites par les objectifs de sécurité), sont représentés par les règles de sécurité. Du point de vue formel, une règle de sécurité est une formule modale mais, contrairement aux objectifs de sécurité, c'est une formule dont les différentes clauses ne sont pas toutes des sous-formules modales, par exemple $p \Rightarrow \mathbf{P}q$. Dans ce cas, la formule précise une relation logique entre l'état existant dans un monde et les règles déontiques qui s'appliquent. Ces règles sont en relation étroites avec les éléments de base de la politique de sécurité qui sont des éléments de sécurité (comme des labels, des rôles, etc.). Elles reflètent la manière dont l'état de sécurité est en relation avec les différentes permissions, interdictions ou obligations qui existent dans le système.

La cohérence entre les différentes règles de sécurité et les règles de fonctionnement du système est une propriété importante qui devrait être assurée dans la spécification (cf 2.1.4). En effet, une incohérence signifie alors qu'il existe des cas de figure dans lesquels, directement ou en raison du fonctionnement du système, deux règles de sécurité conflictuelles sont activées. Ceci signifie que les règles de sécurité choisies, appliquées au système considéré, ne permettent pas toujours de définir un état de sécurité du système, en dehors de l'état vide dans lequel tous les éléments de sécurité ont disparu. En pratique, ce cas de figure peut toutefois éventuellement

1. En effet, on a:
$D = (\mathbf{O}p \Rightarrow \mathbf{P}p) \Leftrightarrow \neg\mathbf{O}p \vee \mathbf{P}p \Leftrightarrow \neg\mathbf{O}p \vee \neg\mathbf{O}\neg p \Leftrightarrow \neg(\mathbf{O}p \wedge \mathbf{O}\neg p) \Leftrightarrow \neg(\mathbf{O}p \wedge \mathbf{F}p) = \neg(8)$

apparaître dans certaines organisations. Il est alors vraisemblable qu'il soit associé à des situations très particulières, voire invraisemblables (par exemple, une situation où un seul individu est associé à tous les rôles existants). Dans un tel cas, on peut éventuellement envisager de tolérer une incohérence de la politique de sécurité. Néanmoins, il peut être tout aussi simple de modifier (en la précisant) la description du fonctionnement dans la spécification de manière à éliminer l'incohérence, sans pour autant trahir les procédures effectivement mises en œuvre dans l'organisation.

En revanche, le problème de savoir si les différentes règles de sécurité sont effectivement compatibles avec les objectifs de sécurité et s'il n'existe pas un moyen, compte tenu de ces règles et des règles de fonctionnement, d'atteindre un état de sécurité impliquant des propriétés modales en contradiction avec les propriétés de sécurité attendues, est un problème distinct qui correspond à la vérification de la politique de sécurité et de ses mécanismes. Dans le cas général, ce problème peut être très délicat à traiter. La spécification de la politique de sécurité permet d'aborder ce problème de vérification du point de vue de la démonstration formelle, mais nous nous intéressons avant tout dans ce mémoire à la possibilité de spécifier des propriétés générales et des mécanismes de sécurité dans le langage déontique.

2.2.4.4 Définitions complémentaires

Nous présentons dans ce chapitre quelques définitions complémentaires qui seront utilisées par la suite. Ces définitions ont trait à la représentation formelle de deux notions: la notion de privilège, et la notion de vulnérabilité.

2.2.4.4.1 Notion de privilège

Un **droit** d_p vis-à-vis de la proposition p au sens de la sécurité est défini par (11) comme la permission de p. Un droit peut donc être vrai ou faux, soit dans l'ensemble du modèle M, soit dans un monde particulier w de M. On note d_p^w le fait que le droit d_p soit vrai dans le monde w.

$$d_p = \mathbf{P}p \tag{11}$$

On définit ensuite un **privilège**, noté P_w comme un ensemble de droits $d_{p_i}^w$ vrais dans un monde w donné (12).

$$P_w = \left\{ d_{p_1}^w, d_{p_2}^w, \ldots, d_{p_n}^w \right\} \tag{12}$$

2.2.4.4.2 Notion de vulnérabilité

Une **vulnérabilité** est définie comme une règle permettant de déduire que, si une certaine propriété non-modale et un droit sont vérifiés dans un monde w, il existe un autre monde w' en relation avec w tel qu'un autre droit soit vrai dans w'. Toutefois, étant donné qu'une distinction semble indispensable entre la notion de permission (directement associée à un droit) et les possibilités offertes par une vulnérabilité (qui ne relèvent pas directement du domaine déontique), l'opérateur **P**

ne semble pas approprié pour représenter une vulnérabilité. Par contre, l'opérateur ontique $\Diamond$ désignant la possibilité correspond à la formulation naturelle. Une vulnérabilité est donc représenté par une règle de la forme de (13).

$$p \wedge d_q \Rightarrow \Diamond d_r \tag{13}$$

Étant donné la définition d'un droit, une vulnérabilité est donc une règle de la forme (14).

$$p \wedge \mathbf{P}q \Rightarrow \Diamond \mathbf{P}r \tag{14}$$

L'introduction simultanée des modalités déontiques (**O**, **P**, et **F**) et ontiques ($\Box$ et $\Diamond$) dans le langage utilisé pour représenter une vulnérabilité conduit à se poser la question des relations que l'on peut établir entre ces différents opérateurs. Ces relations sont décrites par des axiomes d'interaction (cf annexe A, page 159). Dans notre cas, il semble raisonnable d'inclure l'axiome (15) permettant de relier les notions de permission et de possibilité. Ceci permet notamment de déduire de certaines relations de transfert de droits (comme la délégation de pouvoir, correspondant à une formule du type $p \wedge \mathbf{P}q \Rightarrow \mathbf{PP}r$) l'existence d'une vulnérabilité de la forme (14).

$$\mathbf{P}p \Rightarrow \Diamond p \tag{15}$$

2.2.4.5 Le problème de la vérification

Dans cette section, nous donnons un aperçu informel des différents aspects que peut revêtir la vérification de la spécification déontique d'une politique de sécurité. Dans un premier temps, nous détaillons certaines des propriétés formelles qui peuvent être attendues pour la spécification, et qui correspondent à un souci général vis-à-vis de la politique de sécurité. Ensuite, nous énumérons les différentes approches actuellement disponibles pour exploiter la spécification à des fins de démonstration. Ces approches correspondent aux différents calculs et méthodes de résolution utilisables dans ces logiques. Plutôt que de présenter formellement ces techniques, nous nous intéressons avant tout à leur application à l'étude d'une politique de sécurité.

2.2.4.5.1 Propriétés attendues

Dans le cas d'une politique de sécurité, les propriétés que l'on peut vouloir imposer à la spécification sont associées aux problèmes de cohérence détaillés au 2.1.4, et à des propriétés générales qui peuvent influer sur le choix du système modal particulier utilisé.

On peut tout d'abord souhaiter assurer la cohérence des objectifs de sécurité, ce qui revient à exclure les cas où une obligation et une interdiction sont en conflit, c'est-à-dire à vérifier que (16) est vraie pour la spécification effectuée[1].

$$\neg(\mathbf{O}p \wedge \mathbf{F}p) \tag{16}$$

1. Étant donné notre définition de l'opérateur **F**, (16) correspond également à $\neg(\mathbf{O}p \wedge \mathbf{O}\neg p)$.

Ensuite, il peut être désirable d'associer D (page 67) à la spécification. Du point de vue de la sécurité, D signifie qu'une obligation doit impliquer la permission correspondante. Du point de vue de la structure du modèle M associé, D signifie que la relation R possède une propriété particulière: $\forall w, \exists w' | w \mathrm{R} w'$. Par définition, la relation R entre les mondes est qualifiée de *sérielle* [Chellas 1980, p. 80]. Ceci signifie également qu'il n'existe pas, dans la classe des modèles sériels[1], de mondes correspondant à un état de sécurité figé.

On peut également désirer que, lorsqu'une propriété p est obligatoire dans un monde, soit $\models_w \mathbf{O}p$, cette propriété soit également vraie dans ce monde, soit $\models_w p$. Ceci correspond à (17). Ceci correspond à une vision rigide de la sécurité, et du point de vue du modèle sémantique $\mathrm{M} = \langle \mathrm{W}, \mathrm{R}, \mathrm{V} \rangle$ signifie que R est réflexive.

$$\mathbf{O}p \Rightarrow p \tag{17}$$

Il peut être également souhaitable d'imposer (18). Dans ce cas, une obligation dans un monde est propagée à tous les mondes possibles. La relation R est donc transitive, et les propriétés de sécurité énoncées dans une spécification incluant (18) sont donc des propriétés extrêmement fortes.

$$\mathbf{O}p \Rightarrow \mathbf{OO}p \tag{18}$$

Toutes ces différentes propriétés correspondent, du point de vue logique, au choix d'un système modal [Hughes & Cresswell 1968, Appendix three]. Dans notre cas, nous avons opté pour le système KD, consistant en une logique modale normale incluant l'axiome D. Néanmoins, ce choix peut être contestable du point de vue logique (notamment en raison de la difficulté de définir clairement la notion d'obligation conditionnelle, cf page 68 et [Chellas 1980, §6.5]), et on peut envisager d'effectuer d'autres choix. Dans ce cas, les propriétés attendues pour la spécification pourront être plus complexes.

2.2.4.5.2 Méthodes de vérification

On peut utiliser différentes approches pour effectuer des calculs dans un langage modal. En nous basant sur [Bibel & Eder 1993 ; Fitting 1993 ; Fariñas del Cerro & Herzig 1995], nous distinguons:

- La méthode basée sur les systèmes d'axiomes, aussi appelée calcul de Frege-Hilbert qui consiste à baser les raisonnements sur des règles d'inférences généralement peu nombreuses et sur un ensemble d'axiomes [Bibel & Eder 1993, §2.2]. L'application d'une règle d'inférence aux axiomes permet de dériver une nouvelle formule vraie, jusqu'à la démonstration de la propriété recherchée. Cette méthode est difficilement automatisable, notamment en raison de la difficulté de rechercher parmi toutes les déductions possibles la

1. Par souci de concision, on associe directement au modèle un qualificatif correspondant en fait à une propriété de la relation entre les mondes de ce modèle. Un modèle sériel désigne donc en fait un modèle $\mathrm{M} = \langle \mathrm{W}, \mathrm{R}, \mathrm{V} \rangle$ où R est sérielle.

démonstration d'une propriété particulière. Toutefois, on peut noter que la plupart des systèmes modaux sont généralement présentés sous la forme de systèmes d'axiomes et, parfois, on ne dispose pas d'autre méthode de preuve qu'une méthode axiomatique [Fitting 1993, §1.7].

- La méthode de déduction naturelle se rapproche plus de la manière dont les mathématiciens démontrent les théorèmes. Dans ce type de calcul, chaque dérivation démarre d'hypothèses à partir desquelles de nouvelles formules sont établies. Des règles permettent ensuite de rendre une formule dérivée indépendante des hypothèses qui ont été faites. Dans le cadre de la plupart des logiques modales propositionnelles, on dispose d'un système de déduction naturelle [Fitting 1983; Fitting 1993, §1.8].
- Les méthodes de résolution sont les plus répandues et les plus utilisées. Elles s'appuient sur une mise sous forme clausale des formules permettant d'éliminer progressivement certains éléments de la formule. Il s'agit en général de méthodes de réfutation. Plusieurs méthodes ont été proposées pour les logiques modales, et notamment des méthodes mixtes [Fariñas del Cerro & Herzig 1995, §2.4] combinant la résolution à d'autres approches comme la méthode des tableaux.
- La méthode des tableaux et le calcul des séquents de Gentzen peuvent être vus comme des variantes notationnelles l'un de l'autre. L'idée de la méthode des tableaux consiste, pour prouver une formule f, à faire l'hypothèse $\neg f$ et à dériver une contradiction en scindant successivement f en chacune de ses sous-formules jusqu'à ce que l'on obtienne à la fois une formule et sa négation. Dans le cadre des logiques modales, la méthode des tableaux a été appliquée à un certain nombre de systèmes modaux [Fitting 1983; Fitting 1993, §1.9, §2.4, §3.3]. En outre, elle offre l'avantage d'être une méthode constructive qui produit un modèle correspondant à la formule démontrée.
- Les approches par traduction consistent à effectuer une traduction des différentes formules d'un système modal vers un autre système logique [Fariñas del Cerro & Herzig 1995, §2.2]. L'atout de cette approche consiste dans la possibilité d'utiliser un seul outil de preuve dans une logique de base pour effectuer des démonstrations dans toutes les logiques qui peuvent être traduites vers cette logique de base.

Dans le cadre de l'étude des propriétés d'une politique de sécurité, il nous semble important de faire le choix d'une méthode produisant le maximum d'information sur les raisons de l'échec ou du succès de la démonstration des différentes propriétés. Notamment, dans les cas où un objectif de sécurité n'est pas vérifié étant donné les règles de fonctionnement et les règles de sécurité considérées, il est primordial de pouvoir en expliquer la raison et donc de détailler l'état ou la succession d'états qui est à l'origine de l'échec. En effet, l'étude de ce contre-exemple fournit des informations particulièrement utiles pour envisager d'améliorer la sécurité du système ou de l'organisation, ou pour mieux préciser les objectifs de sécurité. Étant donné ces raisons, notre faveur va à l'utilisation d'une méthode de déduction basée

sur la sémantique comme la méthode des tableaux [Fitting 1988 ; Catach 1991 ; Beckert & Posegga 1995], ou d'une méthode de déduction naturelle. Néanmoins, d'autres approches peuvent être suivies [Cholvy & Cuppens 1997, Appendix].

2.3 Application

Dans cette section, nous étudions les modalités de mise en œuvre de la méthode de spécification d'une politique de sécurité présentée précédemment dans deux cas de figure. Tout d'abord, nous nous intéressons à l'application de cette méthode dans le cadre d'une organisation possédant des besoins en matière de sécurité. Ensuite, nous étudions le cas d'un système informatique d'un point de vue théorique en considérant certaines politiques de sécurité utilisées dans ce domaine et d'un point de vue pratique dans le cadre du système d'exploitation UNIX. Quel que soit le cas de figure étudié, la construction d'une politique de sécurité correspond aux différentes étapes présentées au 2.1, néanmoins, des préoccupations spécifiques apparaissent pour chacun de ces différents systèmes d'information.

2.3.1 Application à une organisation

Dans le cas d'une organisation, il est important d'identifier tout d'abord les différentes informations déjà disponibles dans l'organisation et qui peuvent servir de base à la spécification de la politique de sécurité. En effet, notamment dans le cas d'une organisation de grande taille, la définition complète de cette politique à partir de zéro est une tâche très complexe qui peut impliquer des itérations successives entre la description du fonctionnement et la définition des règles et des objectifs de sécurité. Il est nécessaire de s'appuyer le plus possible sur des représentations déjà existantes de la structure, du fonctionnement, ou des exigences de sécurité de l'organisation afin de permettre la définition de la politique de sécurité de la manière la plus efficace possible. Ces informations servent de support à la définition des éléments de description et des règles de sécurité. Dans une deuxième étape, la formulation des objectifs de sécurité de l'organisation s'appuie sur ces premières descriptions (ainsi que sur une éventuelle définition préexistante des besoins si elle existe), et permet d'obtenir une première version de la politique de sécurité. Cette version pourra ensuite être étudiée et améliorée, mais elle constitue un point de départ indispensable pour l'analyse de la sécurité de l'organisation dans le cadre que nous avons présenté. En effet, si les évolutions ultérieures s'intéressent de plus près à la sémantique de la politique de sécurité, et donc à la pertinence de ce qu'elle signifie du point de vue de la sécurité, la première version se caractérise aussi par un travail de description de l'organisation qui conditionne les possibilités d'évolution ultérieures de l'ensemble de la politique.

2.3.1.1 Utilisation de représentations existantes

Dans une première étape, il importe de rechercher dans l'organisation les différents documents qui pourraient potentiellement être utilisés pour construire la politique de sécurité. En effet, dans la plupart des cas, il existe déjà des éléments définissant les éléments de base de la spécification, comme :

- un organigramme fonctionnel qui précise les différentes fonctions présentes dans l'organisation et définit ces fonctions en terme de tâches effectuées par les individus qui les occupent ;
- un organigramme structurel qui décrit, pour chacune des entités de l'organisation (service, unité de production, agence, etc.) le nombre et la nature des postes existant dans ces entités, ainsi que les individus qui occupent ces postes ; la liste du personnel constitue aussi un moyen d'identifier tous les individus qui interviennent dans l'organisation ;
- des documents de définition des tâches effectuées par les différents individus, qui précisent notamment les conditions d'accomplissement d'une tâche et détaillent les différentes opérations nécessaires à sa réalisation ;
- des manuels d'assistance ou des fiches d'aide qui détaillent, pour chacune des opérations à effectuer, les étapes à respecter et les documents qui sont nécessaires ;
- dans certains cas ou dans certains domaines d'activité, des normes de fonctionnement peuvent s'appliquer à l'organisation, soit pour des raisons techniques (comme dans le domaine de la construction aéronautique), soit pour satisfaire à des exigences de qualité. Ces normes (ainsi que les dossiers constitués par l'organisation pour démontrer le respect de la norme) précisent également un certain nombre de caractéristiques du fonctionnement de l'organisation ;
- une description des flux d'information et plus généralement tous les documents produits par l'application d'une technique d'analyse des besoins de l'organisation, qui montrent comment les différents documents circulent dans l'organisation et permettent également de savoir comment les opérations sont reliées entre elles.

Du point de vue de la sécurité, on peut également noter que bon nombre de documents peuvent servir de point de départ à la compréhension des différents mécanismes de sécurité et à l'identification des objectifs de sécurité, par exemple :

- un document identifiant le partage des différentes responsabilités ou un organigramme hiérarchique, qui précisent les pouvoirs des individus, ainsi que les relations hiérarchiques entre les individus ;
- une liste des procédures obligatoires, qui précisent les différentes étapes qui doivent être suivies pour la réalisation d'une opération, si cette opération a un caractère sensible ;
- un règlement interne ou des statuts qui définissent le fonctionnement général théorique de l'organisation ;

- certains textes de lois peuvent réglementer le fonctionnement d'un certain type d'organisation, comme c'est le cas dans le domaine des assurances, dans le domaine financier, ou dans le domaine de la santé; ils précisent des règles impératives que l'organisation doit respecter;
- les dossiers élaborés dans le cadre d'une procédure d'évaluation de la sécurité selon des critères normalisés (TCSEC, ou ITSEC, cf 1.3.1, page 38);
- les résultats d'une analyse des risques préalablement effectuée;
- ou bien sûr un règlement de sécurité, qui est la version en langage naturel d'une politique de sécurité de l'organisation.

Cette liste ne se veut pas exhaustive. Dans la plupart des cas, la culture de l'organisation et son domaine d'activité influent sur la dénomination des documents, sur leur contenu, sur leur structure ou leur niveau de précision. Néanmoins, toute organisation produit ce type de document pour faciliter son organisation interne et son fonctionnement, ou par obligation légale ou technique. En s'appuyant par exemple sur un organigramme, une description structurelle et un règlement de sécurité, il est possible de rassembler la plupart des éléments nécessaires à la spécification d'une politique de sécurité telle qu'elle a été présentée au 2.1. En effet, on peut alors identifier les éléments de base de description de l'organisation, préciser un certain nombre de règles de fonctionnement générales, et même éventuellement identifier des règles de sécurité explicites et des objectifs de sécurité. Dans le paragraphe suivant, nous prenons l'exemple d'une représentation partielle du fonctionnement d'une organisation, basée sur une technique générale de définition structurée des besoins, pour montrer comment cette représentation peut servir de point de départ à notre approche. Cette technique, appelée SRD pour *Structured Requirements Definition* [Orr 1981; Davis 1993], fournit un grand nombre des informations nécessaires à la spécification de la politique de sécurité, mais son domaine d'application naturel n'est pas spécifique à la sécurité.

2.3.1.1.1 Description SRD

La méthode de description SRD est une technique de description structurée des flux d'information présents dans un système. Cette méthode consiste:

- d'abord à construire un diagramme représentant les flux d'information complets de l'organisation, à partir de diagrammes qui définissent la vision de chacun des différents intervenants préalablement interrogés;
- et ensuite à regrouper certains des nœuds de ce diagramme pour pouvoir les agréger et disposer de plusieurs niveaux de représentation du diagramme général.

Une présentation plus complète des différentes étapes de cette méthode pourra être trouvée dans [Davis 1993, pp.78-86], en même temps qu'une présentation générale d'autres techniques de description utilisées dans l'industrie. La figure 11 montre une représentation SRD tirée d'un cas réel décrivant le fonctionnement des achats d'une organisation de taille moyenne, dans lequel deux regroupements ont été effectués (correspondant au service des achats, et à l'unité considérée).

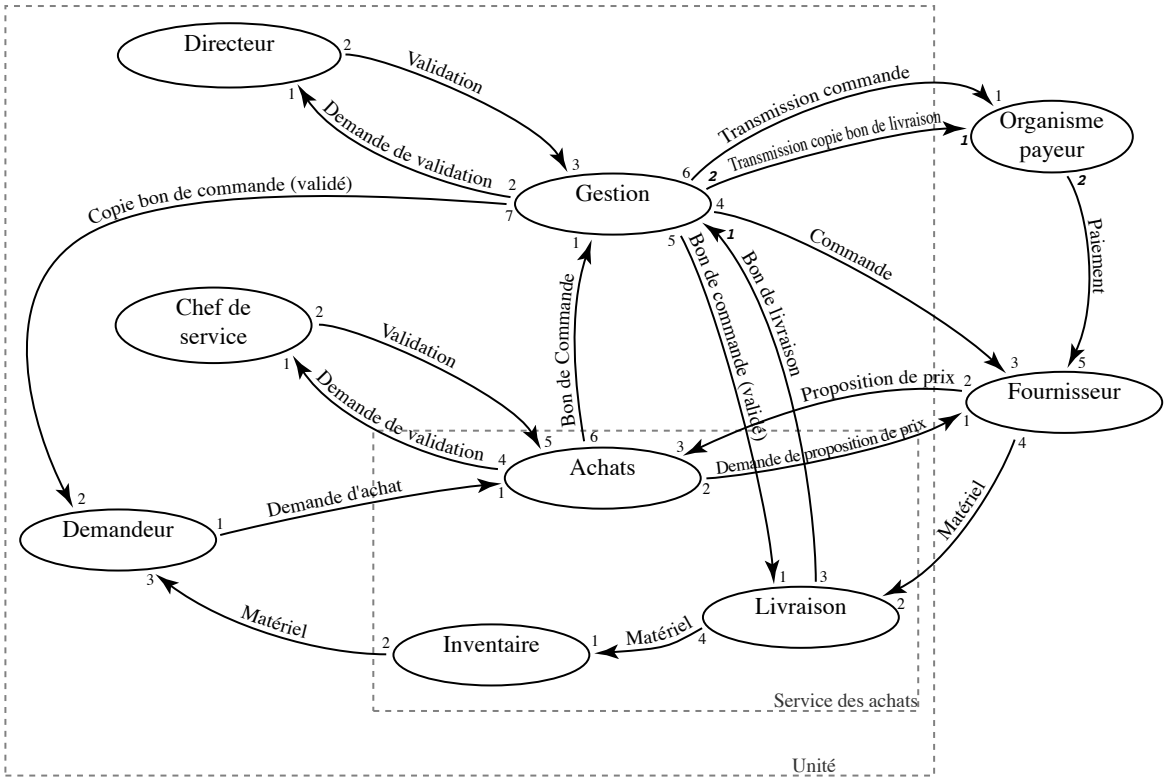


Figure 11 - Représentation SRD d'un service d'achat

Dans une telle description, on identifie aisément un certain nombre des éléments nécessaires à la spécification de la politique de sécurité:

- Les labels des nœuds du diagramme SRD peuvent représenter des éléments de base de la politique de sécurité désignant des rôles (ou éventuellement des individus).
- Les labels des arcs de ce diagramme représentent: soit des documents ou des objets qui transitent d'un nœud à un autre (par exemple une commande ou un matériel), soit différentes actions nécessaires pour que le processus fonctionne. On identifie donc ici de nouveaux éléments de base de la description.
- La structure du diagramme permet d'identifier certaines des règles de fonctionnement de l'organisation qui relient entre elles les opérations représentées dans le diagramme. Par exemple, on note dans la figure 11 que l'émission d'un bon de commande dépend d'une demande d'achat d'un membre de l'organisation et d'une proposition de prix d'un fournisseur. Les arcs d'un diagramme SRD définitif sont numérotés de manière à connaître exactement l'ordre dans lequel sont effectuées les opérations. Cet ordre doit définir complètement les dépendances causales entre les différentes opérations et donc les règles de fonctionnement de l'organisation.

La plupart des descriptions structurées du fonctionnement d'un système ou d'une organisation permettent d'obtenir toutes ces informations. Une description SRD est relativement simple et se prête bien à notre analyse, mais d'autres descriptions (comme SADT, SASS [Ross 1977 ; DeMarco 1979]) permettraient également de rassembler les mêmes éléments.

2.3.1.2 Obtention des règles de fonctionnement et des éléments de base

L'utilisation de représentations existantes permet d'avoir une première idée des différents éléments de description utilisables dans la spécification, ainsi que de leur structure. La vocation de cette partie de la spécification de la politique de sécurité est de servir de support à la formalisation des règles et des objectifs. Dans cette optique on peut également énoncer des règles assez générales concernant la manière de structurer les éléments de description, et les différents éléments dont l'étude est à privilégier.

2.3.1.2.1 Éléments de base

Vis-à-vis de la sécurité, on s'intéresse à la réglementation des différentes opérations réalisées par les individus dans l'organisation, ainsi qu'aux objets qui peuvent servir de support à ces opérations, ou qui en sont la cible. Ainsi, dans la politique de sécurité, on doit habituellement distinguer entre : les individus, les opérations, et les objets. Ces différentes catégories correspondent à celles généralement identifiées dans les différents modèles de sécurité utilisés en informatique. Toutefois, dans une organisation, les objets manipulés par un individu dans le cadre de son métier peuvent être complexes et surtout avoir des caractéristiques très variées (ordinateurs, factures, commandes, contrats, produits commerciaux, argent, etc.). En revanche, on peut généralement distinguer des catégories d'opérations. Ce sont donc les différentes opérations effectuées qui permettent généralement de distinguer de manière assez synthétique les actions d'un individu, et on s'intéressera avant tout à elles. Il est ensuite possible d'essayer de préciser sur quels objets portent ces opérations si c'est utile, mais c'est un raffinement qui peut être complexe. Considérer seulement les opérations permet déjà de structurer les éléments de description des actions effectuées dans le système. Par exemple, dans une agence bancaire, on distinguera les opérations de gestion courante, les opérations de crédit, les souscriptions d'assurance-vie, etc. Chacune de ces catégories peut ensuite être raffinée pour décrire les opérations effectivement effectuées par les individus (par exemple : ouverture de compte, retraits, dépôts, prêt à la consommation, prêt-habitat, opérations nominatives, opérations anonymes).

L'identification explicite de tous les individus et de toutes les opérations existant dans une organisation peut conduire à une liste relativement longue d'éléments. L'introduction d'éléments structurants dans cette énumération s'impose donc. On distingue plusieurs types de regroupement. Des groupes peuvent être utilisés pour rassembler des individus possédant une caractéristique commune (comme l'affectation géographique, le service de rattachement, etc.). Des rôles peuvent être utilisés

pour rassembler différentes opérations rattachées à une même fonction (comme les fonctions de directeur, de commercial, d'agent administratif). Plus généralement, la structure de la description des éléments de base de la politique de sécurité doit refléter assez fidèlement la structure de l'organisation. De plus, cette structure doit faire apparaître les éléments les plus commodes pour exprimer les besoins de sécurité.

La vocation de cette description est de fournir les moyens d'exprimer les besoins de sécurité. Il n'est donc pas forcément nécessaire de détailler l'ensemble des éléments de base, et on peut envisager de rester à un niveau relativement abstrait. Par exemple, dans les cas où les opérations effectuées par l'organisation sont extrêmement nombreuses, il est parfaitement envisageable de ne considérer dans un premier temps que les catégories d'opérations. Si la définition ultérieure des objectifs de sécurité impose de raffiner la structure en précisant le contenu de certaines catégories on envisagera alors de rajouter des éléments de base. Toutefois, ce raffinement sera probablement ponctuel. L'intérêt de maintenir un certain niveau d'abstraction est de diminuer le nombre d'éléments de base inclus dans la spécification (surtout quand ils sont similaires du point de vue de la sécurité). Ceci est évidemment souhaitable, et parfaitement possible. Par exemple, dans une organisation où les besoins de confidentialité ne s'expriment que vis-à-vis des individus extérieurs à l'organisation, il est généralement inutile de distinguer les différentes opérations de consultation des données pour exprimer ce besoin. On peut se contenter de caractériser les conditions générales d'accès aux informations sous deux angles seulement.

2.3.1.2.2 Mécanismes et fonctionnement de l'organisation

L'objectif de la spécification des règles de fonctionnement et des règles de sécurité dans la politique est de préciser l'enchaînement des différentes opérations dans l'organisation. Ainsi que nous l'avons vu dans l'exemple d'une description SRD (cf 2.3.1.1.1) ceci définit de manière très générale les différents flux d'information et les contrôles de sécurité existant dans l'organisation. Afin d'obtenir une description synthétique des différents flux d'information, les règles doivent s'appuyer sur les éléments structurants définis précédemment. Ceci permet également de raffiner les définitions des rôles et des groupes qui avaient été établies préalablement, en précisant leur contenu si nécessaire.

Il est souhaitable que les règles de fonctionnement prennent en compte l'ensemble des opérations (ou des catégories d'opérations) figurant dans la liste des éléments de base. En effet, il s'agit d'identifier comment s'enchaînent les actions qui apparaissent dans l'organisation afin de pouvoir ultérieurement déterminer l'impact de ce fonctionnement sur les objectifs de sécurité. La description des règles de fonctionnement doit être cohérente pour représenter un fonctionnement réel. Enfin, le souci d'abstraction mentionné précédemment, visant à réduire la taille de la description, est également valable pour la description des règles de fonctionnement dont le nombre doit rester limité.

2.3.1.3 Introduction d'objectifs de sécurité

Une fois la description de l'organisation satisfaisante, il est alors possible d'utiliser l'ensemble des éléments déjà définis pour formuler les objectifs de sécurité. Dans les cas où un règlement de sécurité existait déjà dans l'organisation, celui-ci devra évidemment être transposé dans la spécification. Le langage déontique défini au 2.2.3.1 permet d'exprimer très naturellement les objectifs de sécurité, en s'appuyant sur les opérateurs d'obligation, de permission et d'interdiction. La difficulté de cette tâche réside plutôt dans la formulation de propriétés pertinentes vis-à-vis de la sécurité. En effet, une formule déontique peut représenter des propriétés de sécurité très fortes et on devra veiller à ce que ces différentes propriétés représentent bien les besoins réels de l'organisation et ne soient pas trop générales.

2.3.1.4 Raffinement

Vraisemblablement, dans une première version, les objectifs de sécurité présents dans la politique sont des objectifs très généraux, appuyés sur les abstractions introduites au niveau des éléments de base. Il est probable que ces objectifs soient alors trop restrictifs et facilement invalidés par le fonctionnement de l'organisation, voire en contradiction entre eux dans de nombreux cas particuliers. Ceci peut conduire à l'introduction de nouveaux éléments dans la description, par exemple afin de restreindre la portée d'un objectif de sécurité à une certaine catégorie d'opérations. Éventuellement, on peut vouloir introduire de nouvelles opérations. Dans ce cas, on modifie la manière dont on perçoit un objectif de sécurité et on choisit de détailler le fonctionnement de l'organisation pour préciser sur quelles étapes de ce fonctionnement on veut faire porter les propriétés de sécurité. Une spécification de la politique de sécurité de l'organisation suivant la méthode que nous avons présentée permet donc, dans une certaine mesure, d'adapter la politique de sécurité à l'organisation et on peut envisager des compromis sur les contraintes que l'on impose sur son fonctionnement.

Ceci conduit à reprendre éventuellement la description de l'organisation pour la raffiner, préciser les objectifs de sécurité, et améliorer la politique de sécurité. À ce stade, la spécification de la politique de sécurité entre dans un cycle de vie conventionnel. Elle permet d'analyser (éventuellement à l'aide d'outils) la signification de la sécurité pour l'organisation. Cette analyse peut conduire à une meilleure compréhension des besoins de sécurité, ce qui se répercutera bien évidemment au niveau des objectifs de sécurité par des modifications, ou encore à une meilleure compréhension des problèmes posés par le fonctionnement de l'organisation. Ceci peut donner lieu à une description différente de l'organisation dans la politique de sécurité ou, et c'est évidemment un des objectifs principaux, à une modification du fonctionnement de l'organisation visant à améliorer la sécurité.

2.3.1.5 Apports

La définition d'une politique de sécurité apporte un certain nombre d'atouts à l'organisation pour laquelle elle est conçue. En effet, une spécification de la sécurité de l'organisation est un document de référence essentiel dans le cas d'une mise en cause de l'organisation ou d'un de ses membres, et offre un moyen de guider une éventuelle recherche des vulnérabilités existant dans l'organisation.

2.3.1.5.1 Clarification des responsabilités

Dans le cas d'une défaillance grave vis-à-vis de la sécurité, un problème délicat dans une organisation reste l'identification a posteriori des responsabilités des différents individus participant au fonctionnement. Outre un effort direct de compensation des dommages (comme une modification de stratégie survenant à la suite de la perte d'un élément confidentiel par exemple) le traitement d'une défaillance de sécurité fait généralement intervenir une action disciplinaire ou légale. En effet, nous sommes fréquemment dans le cas où une action intentionnelle est la cause directe ou indirecte de la défaillance et le ou les individus l'ayant provoquée sont susceptibles d'être mis en cause directement. Pourtant, la situation simple dans laquelle un agresseur extérieur malveillant a, par ses seuls moyens, mis en défaut la sécurité du système, n'est généralement pas la plus courante. En effet, c'est probablement en tirant parti de certaines failles dans la sécurité du système, voire en s'adjoignant la participation (volontaire, mais parfois aussi involontaire) d'éléments situés à l'intérieur même de l'organisation, qu'une attaque réussie peut être mise en œuvre. Ainsi, des études récentes montrent que les personnels de l'intérieur de l'entreprise sont plus ou moins impliqués dans 70% des cas [CLUSIF 1997]. Dans d'autres situations, comme c'est par exemple le cas vis-à-vis des données médicales, les dommages provenant d'une défaillance de sécurité peuvent être ressentis à l'extérieur de l'organisation, et l'organisation sera alors la cible d'une enquête visant à déterminer, non seulement si ses mécanismes de sécurité ont été détournés, mais aussi si les objectifs de sécurité qu'elle affiche sont bel et bien assurés, et correspondent bien à ce que l'on peut attendre d'elle dans ce domaine. Étant donné la gravité des situations qui peuvent survenir à la suite d'une défaillance de sécurité, et étant donné la difficulté d'attribuer clairement en cas de défaillance la responsabilité des dommages, un des premiers apports d'une politique de sécurité est justement de servir d'argument pour ce jugement. La spécification formelle que nous avons présentée met en effet clairement en évidence les responsabilités des différents individus présents dans l'organisation, grâce à la notion d'obligation qu'elle introduit explicitement. Par ailleurs, les objectifs de sécurité de l'organisation elle-même sont clairement identifiés, et on peut juger de leur adéquation à la vocation de l'organisation avant qu'une défaillance et les poursuites qui peuvent en découler ne soulèvent ces questions. Enfin, si l'obtention des propriétés de sécurité attendues pose des difficultés de mise en œuvre, les règles de sécurité présentes

dans la politique mettent clairement en évidence les choix qui ont été faits vis-à-vis des compromis généralement inévitables entre l'assurance d'un niveau de sécurité élevé et le fonctionnement efficace de l'organisation.

À chacun de ces niveaux, l'introduction d'une politique de sécurité clarifie la vision de la sécurité telle qu'elle est prise en compte dans l'organisation. Dans la plupart des cas, elle permet d'éviter les ambiguïtés, les interprétations multiples, ce qui facilite bien évidemment son analyse, mais aussi sa compréhension, et l'appréciation de sa pertinence [David 1995]. Du point de vue de tous ceux qui à l'intérieur de l'organisation participent à son fonctionnement, ou à l'extérieur utilisent ses services mais peuvent aussi avoir à juger sa sécurité, une politique de sécurité apporte donc une identification claire des responsabilités supportées et de la confiance qui a été accordée.

2.3.1.5.2 Recherche des vulnérabilités

Indépendamment des atouts que peut apporter en règle générale une formulation explicite de la sécurité pour un système ou une organisation, l'utilisation d'une spécification formelle de la politique de sécurité permet d'obtenir directement un certain nombre d'informations. Notamment, à partir des objectifs de sécurité et des règles décrivant le fonctionnement de l'organisation, il est possible d'identifier les différentes opérations effectuées dans l'organisation qui sont importantes pour la sécurité. En effet, soit ces opérations figurent alors explicitement dans un objectif de sécurité, soit elles sont associées à la réalisation d'une opération qui l'est. Les ensembles d'opérations ainsi identifiés permettent de concentrer l'analyse de la sécurité sur certains aspects du fonctionnement de l'organisation.

Cette identification (si possible systématique) des mécanismes sensibles qui sont mis en œuvre dans l'organisation fournit un premier élément à l'étude des différentes vulnérabilités qui pourraient y être présentes. En effet, dans le cadre d'une recherche des faiblesses de l'organisation, on s'intéressera tout d'abord à l'étude des différentes opérations directement concernées par les objectifs de sécurité, et on cherchera à évaluer la difficulté de les effectuer de manière non autorisée.

Cette approche permet d'étudier les éléments de l'organisation intéressants pour la sécurité sans faire appel à des données préexistantes qui rassembleraient les différents types de vulnérabilités communément identifiées dans une organisation ou un système de même type. Néanmoins, si on dispose d'une base de données de vulnérabilités connues, on l'inclura bien évidemment dans l'analyse de validation de la sécurité du système. Toutefois, l'existence ou la pertinence d'une telle base de données n'est pas assurée. L'approche constructive rendue possible par l'intégration du fonctionnement de l'organisation à la description de ses objectifs de sécurité peut permettre d'effectuer une recherche des différentes vulnérabilités potentiellement présentes dans le système à partir d'une analyse de la spécification de la politique de sécurité. Cette opportunité est largement associée à la possibilité d'effectuer des manipulations dans le langage de la logique déontique, par conséquent en pratique à l'existence d'outils de preuve dans ce domaine de la logique formelle et à leur faci-

lité d'utilisation. C'est une des applications possibles d'une spécification des politiques de sécurité telle que nous l'avons présentée qui mérite certainement des développements supplémentaires.

2.3.1.6 Exemple d'application : une agence bancaire

Afin d'illustrer l'application à une organisation de la méthode de spécification d'une politique de sécurité que nous proposons, nous présentons un exemple de sa mise en œuvre dans un cas pratique. Cet exemple est également présenté dans [Ortalo 1998 ; Ortalo & Deswarte 1998b]. L'organisation considérée est une agence bancaire de taille moyenne, employant une trentaine de personnes. La politique de sécurité présentée dans cette section a été construite à partir de différents documents décrivant l'organisation de cette agence, les fonctions assurées par ses membres et les opérations relatives aux métiers bancaires. L'analyse de ces documents a été complétée par une étude sur le terrain de plusieurs jours. La politique de sécurité présentée dans cette section a été construite à partir de différents documents fournis par l'organisation considérée. Plus précisément, trois documents principaux ont servi de point de départ au travail de spécification :

- un organigramme de l'agence ;
- un document de définition des fonctions assurées au sein de l'organisation qui présente les différents emplois occupés par les agents ;
- et la documentation technique décrivant les différentes opérations bancaires qui est mise à la disposition des différents agents pour les assister.

Ce dernier document identifie plus d'une centaine d'opérations qui peuvent être mises en œuvre quotidiennement dans une banque. De ces nombreuses opérations, seul un nombre réduit est étudié dans le cadre de la spécification que nous présentons, et plus précisément les opérations relatives à l'attribution des crédits et à la manipulation de certains titres aux porteurs. Ces opérations paraissent intéressantes car elles mettent en jeu, respectivement, un mécanisme de délégation de pouvoir et le problème de l'anonymat.

2.3.1.6.1 Eléments de base

Les éléments initiaux de la spécification de la politique de sécurité sont présentés dans la figure 12. (Les éléments qui ne sont pas définis directement dans cette figure sont précisés par la suite dans les figures 13, 14 et 15.) Sont représentés les principaux éléments de description utilisés pour définir les règles et les objectifs de sécurité de la spécification. Dans cet exemple, on considère, d'une part les différents individus concernés par la politique de sécurité et notamment les agents de l'organisation, d'autre part un certain nombre d'actions (définies de manière synthétique). Enfin, on notera l'introduction de différents de rôles correspondant aux différentes fonctions de l'organisation. Ces rôles permettent de préciser la définition des règles de fonctionnement ou des règles de sécurité de manière générique et structurée indépendamment des différents agents présents dans l'organisation. L'affectation

des individus concernés par la politique de sécurité aux différents rôles établit ensuite le lien permettant de représenter les attributions et les actions de chacun dans l'organisation.

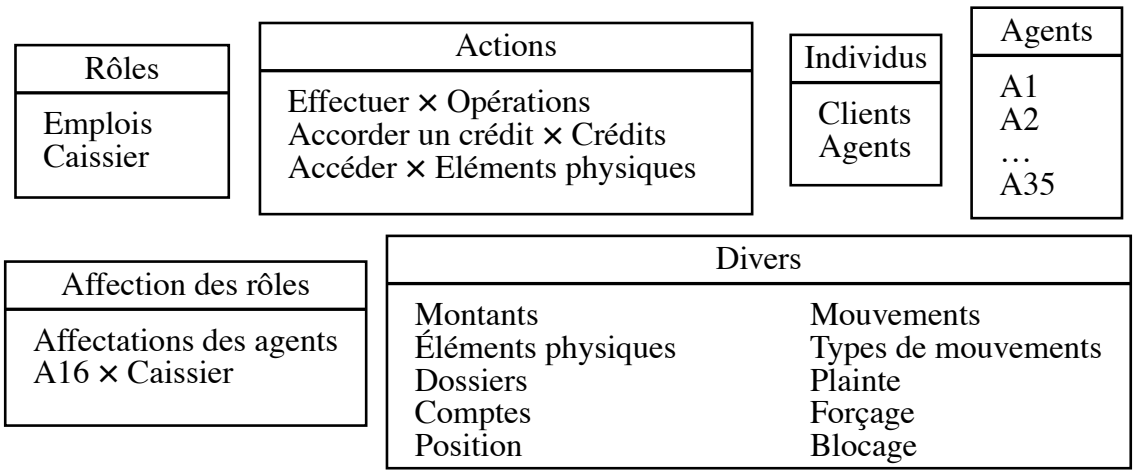


Figure 12 - Racine de la politique de sécurité

La figure 13 détaille les éléments de description utilisés dans la politique de sécurité. On y trouve donc la définition des différentes catégories d'opérations effectuées dans l'organisation, la définition des différentes fonctions existant dans l'organisation qui constituent les différents rôles, et un certain nombre d'éléments complémentaires utilisés dans la spécification (comme certains éléments physiques, la représentation d'informations bancaires générales, etc.). L'affectation précise de chaque agent aux différents rôles est définie dans la figure 14. Enfin, la figure 15 définit les comptes et mouvements bancaires tels qu'ils sont pris en compte (de manière abstraite) dans la politique de sécurité.

2.3.1.6.2 Règles de fonctionnement et règles de sécurité

La figure 16 présente la définition plus précise de certaines des règles de fonctionnement et des règles de sécurité observées dans l'organisation. Du point de vue du fonctionnement, on s'est attaché à caractériser le fait que la quasi-totalité des opérations peuvent être effectuées à partir des terminaux informatiques, ainsi qu'à préciser de manière assez générale l'impact des opérations bancaires courantes sur un compte donné (en se limitant toutefois à préciser les conditions sous lesquelles le solde d'un compte peut être positif ou négatif). On précise également l'action résultant d'un accord de crédit entre la banque et son client. Du point de vue des règles de sécurité, ce fonctionnement est régi par un certain nombre de règles simples. D'une part, tous les agents de l'organisation sont autorisés à effectuer des opérations courantes. En cas de blocage d'une opération (résultant par exemple d'un solde insuffisant sur le compte d'un client), ils sont également autorisés à forcer cette opération, mais celle-ci produit alors une trace spécifique dans les fichiers d'audit repérée par l'étiquette 'Forçage' (ces fichiers sont systématiquement réexa-

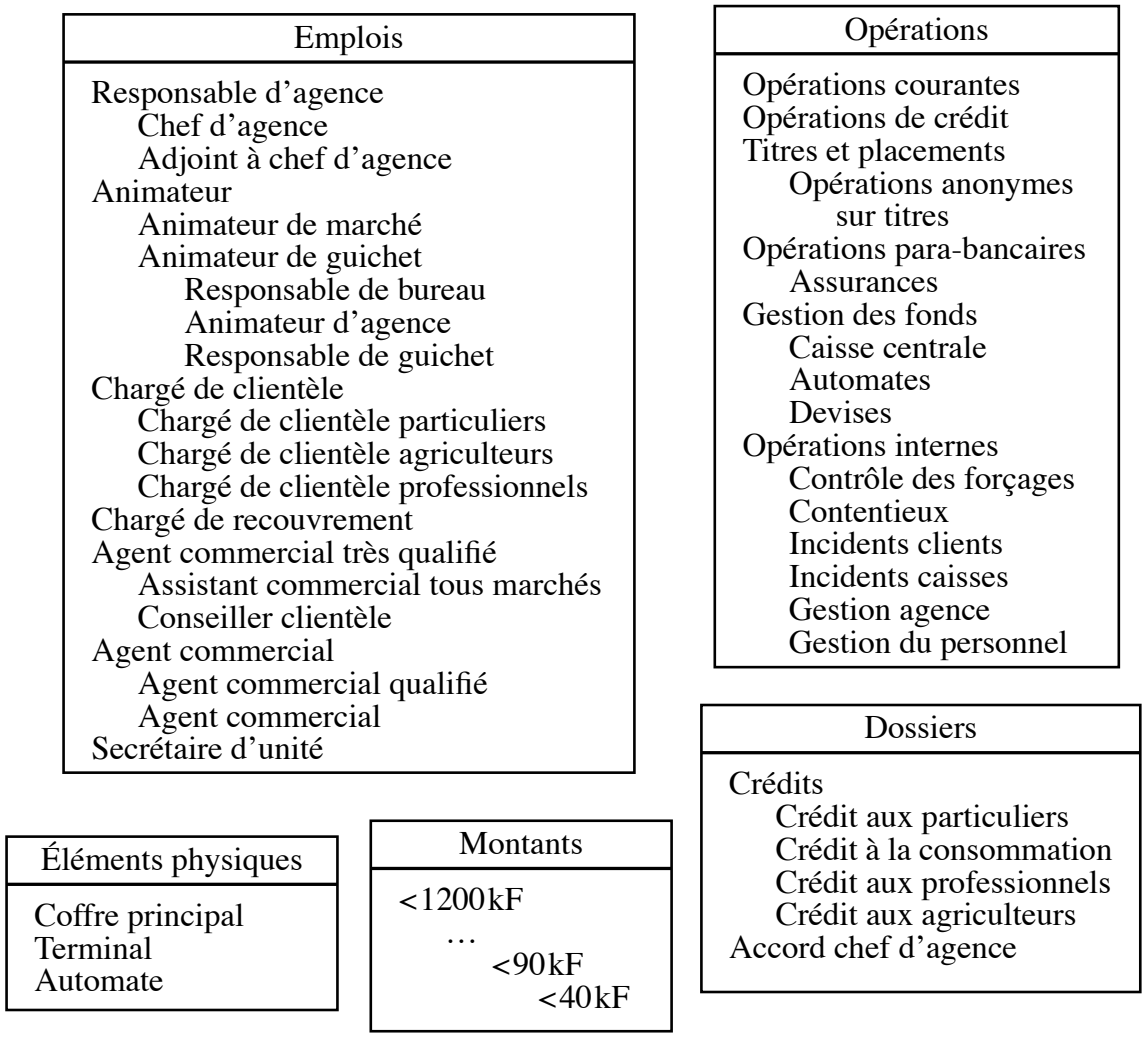


Figure 13 - Eléments de description de la politique de sécurité[a]

a. Dans cette figure, le signe '<' ne désigne pas un opérateur arithmétique. Il est simplement utilisé comme une abréviation pour l'énoncé 'montant inférieur à'. Ceci impose de définir et d'ordonner manuellement tous les montants, et plus généralement tous les nombres, utilisés dans la spécification. Cette restriction peut être éliminée dans la pratique en introduisant des opérateurs et des atomes spéciaux, indépendants du langage déontique, correspondant à l'arithmétique usuelle.

minés par la suite). Enfin, concernant l'accès à un élément physique particulier (le coffre principal de l'agence), une règle de sécurité particulière s'applique pour en restreindre l'accès.

En ce qui concerne les opérations de crédit, si tous les agents sont en mesure de constituer un dossier, l'accord définitif ne fait généralement pas partie de leurs prérogatives. Dans certains cas toutefois, un agent peut être autorisé à accorder directement un crédit. Ces pouvoirs sont attribués par un mécanisme de délégation qui peut être activé par le chef de l'agence moyennant un certain nombre de conditions. Ce mécanisme de délégation est décrit par la règle de délégation présentée dans la

Affectations des agents	
A1 × Chef d'agence	A17 × Agent commercial qualifié
A2 × Adjoint à chef d'agence	A18 × Agent commercial qualifié
A3 × Secrétaire d'unité	A19 × Chargé de clientèle particuliers
A4 × Secrétaire d'unité	A20 × Chargé de clientèle particuliers
A5 × Animateur d'agence	A21 × Conseiller clientèle
A5 × Chargé de clientèle particuliers	A22 × Agent commercial qualifié
A6 × Animateur d'agence	A23 × Conseiller clientèle
A6 × Chargé de clientèle professionnels	A24 × Conseiller clientèle
A6 × Chargé de clientèle agriculteurs	A25 × Agent commercial qualifié
A7 × Responsable de guichet	A26 × Chargé de clientèle particuliers
A7 × Chargé de clientèle professionnels	A27 × Chargé de clientèle particuliers
A8 × Assistant commercial tous marchés	A28 × Chargé de clientèle particuliers
A9 × Chargé de clientèle professionnels	A29 × Chargé de clientèle particuliers
A10 × Chargé de clientèle professionnels	A29 × Responsable de guichet
A11 × Chargé de clientèle professionnels	A30 × Chargé de clientèle particuliers
A12 × Chargé de clientèle agriculteurs	A31 × Chargé de clientèle particuliers
A13 × Chargé de clientèle agriculteurs	A32 × Chargé de clientèle particuliers
A14 × Chargé de clientèle agriculteurs	A33 × Chargé de clientèle particuliers
A15 × Agent commercial qualifié	A34 × Chargé de clientèle particuliers
A16 × Agent commercial qualifié	A35 × Conseiller clientèle

Figure 14 - Éléments de description : affectation d'une fonction à chaque agent

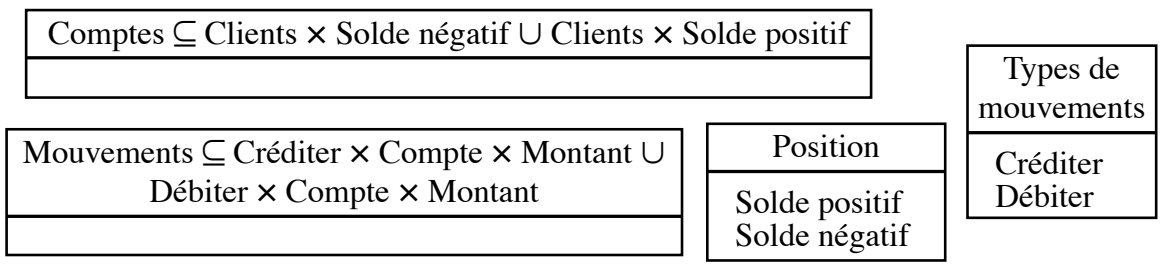


Figure 15 - Éléments de description : représentation des comptes et des mouvements bancaires

figure 17. La figure 17 précise également les conditions qui doivent être vérifiées pour que le chef d'agence puisse accorder une délégation aux agents placés sous sa direction.

2.3.1.6.3 Objectifs de sécurité

La figure 18 présente trois objectifs de sécurité simples qui peuvent être étudiés à partir de cette spécification partielle de la politique de sécurité. Le premier objectif s'applique aux agents même de l'organisation et concerne le crédit. Il s'agit là de garantir que, pour les crédits d'un montant important (cette limite a été choisie *arbitrairement* dans l'exemple présenté), les règles d'autorisation définies dans la figure 17 sont effectivement appliquées. Le deuxième objectif de sécurité vise à

Règles de fonctionnement
Agents × Actions ⇒ Actions Agents × (Effectuer × Opérations) ⇐ Agents × (Accéder × Terminal) Effectuer × Opérations de crédit ⇒ Crédits Effectuer × Opérations courantes ⇒ Débiter × Comptes × Montants ∨ Créditer × Comptes × Montants Client × Solde négatif ⇒ Blocage × (Débiter × Compte × Montants) Accorder un crédit × Crédits ⇒ Créditer × Comptes × Montants

Règles de sécurité
P(Individus × (Effectuer × Opérations courantes)) ⇒ (Individus × Emplois) **P**(Agents × (Effectuer × Opérations de crédit)) ⇒ (Agents × Chargé de clientèle) **P**(Agents × (Accéder × Coffre principal)) ⇒ (Agents × Caissier) ∨ (Agents × Chef d'agence) Agents × Actions ∧ Blocage × Actions ⇒ **O**(Forçage × Agents × Actions) **P**(Délégation chef d'agence) Règle de délégation

Figure 16 - Description des règles de fonctionnement et des règles de sécurité

garantir que les opérations effectuées par les agents sont conformes à l'intérêt des clients. Il s'agit donc de garantir qu'aucun client ne soit en mesure d'émettre une plainte légitime (au sens pénal) contre l'organisation ou un de ses agents. Enfin, le dernier objectif concerne les individus extérieurs à l'organisation et vise à interdire à ces individus la réalisation directe d'opérations courantes. Cet objectif est donné à titre d'exemple et ne sera pas étudié plus à fond par la suite en raison de l'absence de données expérimentales complètes concernant les vulnérabilités affectant les opérations courantes.

La spécification obtenue est une description partielle et d'assez haut niveau de la politique de sécurité de l'organisation considérée. Les opérations bancaires effectuées dans l'agence étudiée sont en effet très nombreuses, et leur analyse exhaustive demanderait un effort très important. Afin d'illustrer la méthode de spécification, nous nous sommes contentés d'étudier certains types d'opérations, qui nous paraissaient intéressants. Une étude approfondie consisterait à raffiner la spécification obtenue pour prendre en compte le fonctionnement exact de chaque type d'opération. Par exemple, pour compléter la description du traitement des opérations courantes, il s'agirait d'intégrer les méthodes d'authentification mises en œuvre au niveau des terminaux informatiques. Il faudrait également raffiner la description des différentes opérations pour isoler celles qui se distinguent (comme le traitement des chèques, les manipulations de matière, ou les opérations de change), et compléter la définition des objectifs de sécurité. Toutefois, la spécification obtenue reflète assez

Règle de délégation
Délégation chef d'agence ∧ (Agents × Emplois) ∧ Autorisations de crédit ⇒ **P**(**P**(Agents × (Accorder un crédit × Crédits)))

Autorisations de crédit
Marché des professionnels Chargé de clientèle × (Crédit aux professionnels × <350kF) Animateur × (Crédit aux professionnels × <450kF) Responsable d'agence × (Crédit aux professionnels × <550kF) Marché des agriculteurs Chargé de clientèle × (Crédit aux agriculteurs × <350kF) Animateur × (Crédit aux agriculteurs × <450kF) Responsable d'agence × (Crédit aux agriculteurs × <550kF) Marché des particuliers Chargé de clientèle × (Crédit aux particuliers × <200kF) Animateur × (Crédit aux particuliers × <450kF) Responsable d'agence × (Crédit aux particuliers × <700kF) Projet de consommation Chargé de clientèle × (Crédit à la consommation × <40kF) Animateur × (Crédit à la consommation × <90kF) Responsable d'agence × (Crédit à la consommation × <140kF) Marché des entreprises Animateur × (Crédit à la consommation × <600kF) Responsable d'agence × (Crédit à la consommation × <1200kF)

Figure 17 - Principe de la délégation pour les autorisations de crédit

Objectifs de sécurité
F(Agents × (Accorder un crédit × Crédits) ∧ (Agents × Emplois) ∧ ¬Autorisations de crédit ∧ Crédits × ¬(<500kF)) **F**(Clients × Plainte) **F**(Clients × (Effectuer × Opérations courantes))

Figure 18 - Objectifs de sécurité

fidèlement l'organisation générale de l'agence, et, ainsi que nous le verrons par la suite (cf 3.2.2) permet l'étude de plusieurs objectifs de sécurité correspondant à des opérations particulières.

2.3.2 Modélisation de la sécurité informatique

Nous présentons dans ce chapitre quelques applications du langage logique $\mathcal{L}_{\mathbf{O}}$ dans le cadre de la modélisation de la sécurité d'un système informatique. Nous nous intéresserons donc à la transposition dans ce langage de quelques uns des

modèles de sécurité et des politiques de sécurité présentés au 1.2.2 (page 14). L'objectif de cette reformulation est de montrer que le langage de spécification que nous proposons est suffisamment général pour permettre de représenter certains des modèles de sécurité communément utilisés, ce qui correspond à notre souhait de proposer une méthode de spécification aussi peu restrictive que possible.

2.3.2.1 Représentation d'un schéma d'autorisation

La définition des politiques d'autorisation d'un système informatique s'appuie souvent sur des modèles généraux comme ceux basés sur les matrices de contrôle d'accès. Nous reprenons ici quelques uns des modèles présentés au 1.2.4.1, page 17, basés sur cette notion, afin de montrer comment il est possible de les représenter dans le langage de spécification $\mathcal{L}_{\mathbf{O}}$.

2.3.2.1.1 Matrice de contrôle d'accès

Les éléments introduits par un modèle basé sur une matrice de contrôle d'accès (cf 1.2.4.1, page 17) correspondent à:

- un ensemble $S = \{s_1, s_2, \ldots, s_n\}$ de sujets;
- un ensemble $O = S \cup \{o_1, o_2, \ldots, o_m\}$ d'objets (contenant S);
- et un ensemble D de droits d'accès qui correspondent aux différentes opérations existant dans le système. Plutôt que d'énumérer directement ces droits, nous choisissons ici de représenter ces différentes opérations par l'ensemble $T = \{t_1, t_2, \ldots, t_l\}$, et donc $D = \{d_{t_1}, d_{t_2}, \ldots, d_{t_l}\}$.

Une action effectuée par le système implique l'association d'un sujet, d'un objet, et d'une opération. Les différentes actions existant dans le système sont donc décrites par les éléments de l'ensemble $E = S \times T \times O$. La matrice de contrôle d'accès précise, pour chaque sujet et chaque objet, l'ensemble des droits d'accès que ce sujet possède sur un objet à un instant donné. Elle correspond donc à la définition d'un ensemble d'autorisations $A \subseteq S \times O \times D$. L'implémentation du système est censée mettre en œuvre une règle de fonctionnement stricte qui consiste à ne permettre la réalisation d'une opération par un sujet sur un objet que dans le cas où l'autorisation correspondante est explicitement mentionnée dans la matrice. Le fonctionnement du contrôle d'accès basé sur cette matrice correspond donc à la règle (19) c'est-à-dire au fait que l'exécution de l'opération t pour le compte du sujet s sur l'objet o est conditionnée par la présence du droit d_t dans la case (s, o) de la matrice de contrôle d'accès A.

$$s \times t \times o \Rightarrow s \times o \times d_t \tag{19}$$

On peut noter que, dans ce cas, un droit d'accès d_t ne correspond pas directement à la notion de droit telle qu'elle a été introduite au 2.2.4.4.1, qui fait intervenir un opérateur modal. En effet, on considère ici que le lien entre l'existence d'un droit d'accès dans la matrice et la réalisation d'une action est fait au niveau du fonctionnement du système. Le modèle basé sur la notion de matrice d'accès ne fait donc pas vraiment de distinction (du point de vue de la sécurité) entre un droit d'accès et

les opérations correspondantes. Il est pourtant possible de donner une dimension déontique à la description de ce fonctionnement, en énonçant, à la place de la règle de fonctionnement (19), la règle de sécurité (20). Cette règle signifie alors qu'un sujet doit posséder un droit d'accès sur un objet pour qu'il lui soit permis d'effectuer l'opération correspondante sur celui-ci.

$$\mathbf{P}(s \times t \times o) \Rightarrow s \times o \times d_t \tag{20}$$

Cette formalisation représente le modèle basé sur les matrices de contrôle d'accès dans son cas le plus général, c'est-à-dire dans le cas où une opération peut avoir un impact sur les permissions et modifier le contenu de la matrice. Dans les évolutions ultérieures de ce modèle, le modèle HRU (cf 1.2.4.1.1, page 17) ou le modèle *Take-Grant* (cf 1.2.4.1.2, page 19), les opérations permettant de modifier la matrice sont d'un type particulier, et sont donc associées à de nouvelles règles de sécurité spécifiques. Dans ce cas, l'utilisation de (19) permet de représenter le fonctionnement général de l'autorisation du système quand celui-ci n'implique pas de modifications de la matrice, et des règles de sécurité définissent les conditions dans lesquelles la matrice peut être modifiée.

2.3.2.1.2 Règles de cession de droits

Les opérations de modification de la matrice (ou commandes) du modèle HRU (tableau 1, page 18) correspondent à l'introduction de règles de sécurité qui contrôlent la manière dont de nouveaux droits d'accès peuvent être attribués. Ces règles, qui doivent être de la forme de (21), signifient que, moyennant une certaine condition portant sur des droits d'accès existants dans un monde, il est permis d'atteindre un monde possible dans lequel de nouveaux droits d'accès ont été attribués ou retirés, ou de nouveaux sujets ou objets ont été créés ou détruits.

$$C = s' \times o' \times d_t' \wedge s'' \times o'' \times d_t'' \wedge \ldots \wedge s^{(m)} \times o^{(m)} \times d_t^{(m)}$$

$$\begin{cases} C \Rightarrow \mathbf{P}(s \times o \times d_t) \\ C \Rightarrow \mathbf{P}(\text{Créer} \times s) \\ C \Rightarrow \mathbf{P}(\text{Créer} \times o) \\ C \Rightarrow \mathbf{P}(\neg(s \times o \times d_t)) \\ C \Rightarrow \mathbf{P}(\text{Détruire} \times s) \\ C \Rightarrow \mathbf{P}(\text{Détruire} \times o) \end{cases} \tag{21}$$

Dans le cadre du modèle *Take-Grant* (cf 1.2.4.1.2, page 19) on distingue parmi les différents types d'opérations de T les types *grant* et *take*. En dehors de la création d'objets ou du retrait des droits qu'un sujet peut effectuer sur des objets qu'il crée ou auxquels il a déjà accès (opérations *create* et *remove*), il n'est possible de modifier l'état de la matrice des droits d'accès qu'en s'appuyant sur les opérations particulières *grant* et *take*. Les règles régissant la modification des droits d'accès dans la

matrice, c'est-à-dire les règles de réécriture du graphe utilisé comme représentation (figure 3, page 20), sont donc relatives à ces manipulations, et sont retranscrites par (22).

$$\forall t \in T \quad \begin{cases} s \Rightarrow \mathbf{P}(\text{Créer} \times o \wedge (s \times o \times d_t)) \\ s \times o \times d_t \Rightarrow \mathbf{P}(\neg(s \times o \times d_t)) \\ (s \times o \times d_t) \wedge (s \times s' \times d_{grant}) \Rightarrow \mathbf{P}(s' \times o \times d_t) \\ (s \times o \times d_t) \wedge (s' \times s \times d_{take}) \Rightarrow \mathbf{P}(s' \times o \times d_t) \end{cases} \tag{22}$$

2.3.2.2 Représentation d'une politique de sécurité

2.3.2.2.1 Propriétés d'une politique multi-niveaux

Afin de représenter les propriétés d'une politique multi-niveaux du type de la politique de Bell-LaPadula (cf 1.2.5.1, page 23), il est tout d'abord nécessaire d'introduire la notion de niveau, et d'ordonner ces niveaux. Tout d'abord, on définit dans (23) un ensemble *Cl* de classifications, et on définit la relation d'ordre total entre ces classifications en ajoutant au langage un certain nombre de faits décrivant cette relation. Ceci correspond en fait à une définition *in extenso* de la relation d'ordre (on énumère toutes les relations possibles). Étant donné le faible nombre de classifications utilisées, ceci ne pose pas de difficultés.

$$Cl = \{\text{Non-Classifié, Confidentiel, Secret, Très-Secret}\}$$

Non-Classifié × Très-Secret	Confidentiel × Très-Secret	Secret × Très-Secret	Très-Secret × Très-Secret
Non-Classifié × Secret	Confidentiel × Secret	Secret × Secret	
Non-Classifié × Confidentiel	Confidentiel × Confidentiel		
Non-Classifié × Non-Classifié			

(23)

Ensuite, il est nécessaire de définir les différentes catégories permettant de répartir l'information (et les niveaux de sécurité correspondant) selon son domaine. L'ensemble *Ca* des catégories étant donné, un compartiment est associé à plusieurs catégories, ainsi qu'indiqué dans la figure 19.

L'ensemble *N* des niveaux de sécurité de la politique multi-niveaux est alors le produit cartésien de l'ensemble des classifications et de l'ensemble des compartiments. Un niveau de sécurité *n* correspond donc à une classification et un compartiment. Une relation entre les niveaux est donnée par (24) où *n* désigne un niveau, *cl* une classification, *Co* un compartiment et *cat* une catégorie, et correspond à la relation d'ordre partiel (ou relation de dominance) notée $\preceq$ (cf 1.2.5.1, page 23). Les différents niveaux forment donc un treillis partiellement ordonné.

$$n \preceq n' = n \times n' = (cl \times Co) \times (cl' \times Co') \Leftrightarrow cl \times cl' \wedge (Co \times cat \Rightarrow Co' \times cat) \tag{24}$$

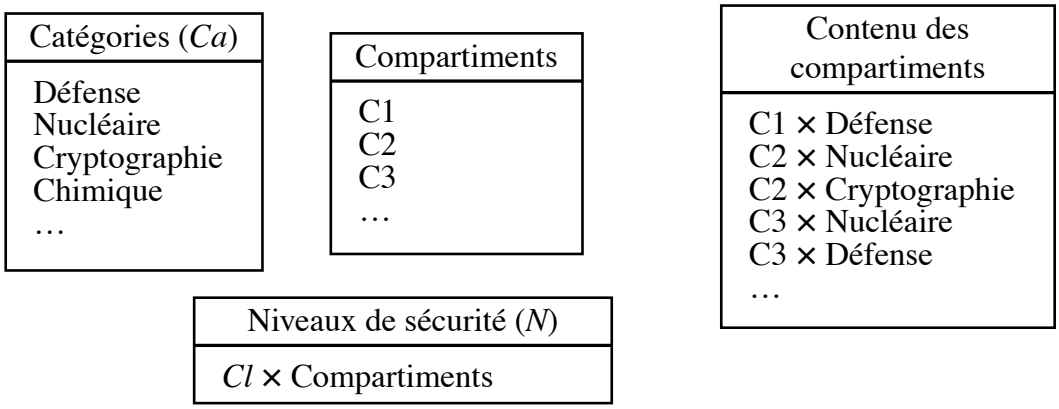


Figure 19 - Définition des catégories, des compartiments et des niveaux de sécurité

Chacun des sujets et des objets pris en compte dans la politique de sécurité doit être associé à un niveau par une relation, notée $s \times n$ ou $o \times n$. Par ailleurs, on considère, comme précédemment, deux types d'opérations : *lecture* et *écriture*. Les droits d'accès correspondant sont notés a_{lecture} et $a_{\text{écriture}}$.

La définition des deux règles de sécurité de la politique de Bell-LaPadula est donnée par (25). On notera que ces deux règles ne font pas intervenir d'opérateur déontique. Il s'agit donc en fait de règles de fonctionnement qui précisent la manière dont le système doit effectuer les contrôles d'accès pour chaque opération.

$$\begin{cases} s \times o \times a_{\text{lecture}} \Rightarrow s \times n_s \wedge o \times n_o \wedge n_o \times n_s \\ s \times o \times a_{\text{lecture}} \wedge s \times o' \times a_{\text{écriture}} \Rightarrow o \times n \wedge o' \times n' \wedge n \times n' \end{cases} \quad (25)$$

2.3.2.2.2 Notion de capacité

L'inutilité d'utiliser des opérateurs déontiques pour représenter les règles de la politique de Bell-LaPadula provient du fait que dans le modèle de sécurité associé, tout comme dans le modèle de Lampson, les droits d'accès présents dans la matrice représentent directement les notions de permission et d'obligation. Il en résulte que, pour qu'un système vérifie les propriétés définies dans ces modèles, chaque opération qu'il effectue doit impliquer une vérification de ces droits d'accès. Cette consultation systématique de la matrice pose des problèmes délicats de performance pour leur implémentation, notamment dans le contexte des systèmes distribués. Une solution à ces problèmes, présentée dans [Nicomette 1996], consiste à considérer les droits d'accès présents dans la matrice comme des droits symboliques. Dans ce schéma d'autorisation, la réalisation de toutes les opérations qui existent dans le système ne nécessite pas forcément l'existence d'un tel droit dans la matrice. En revanche, pour effectuer une opération, un sujet doit présenter une capacité pour cette opération. Quand un sujet initie la réalisation d'une opération complexe, impliquant la mise en oeuvre d'autres opérations, on effectue une vérification de l'existence du droit symbolique correspondant dans la matrice. Ensuite on génère,

grâce à des règles spécifiques, une ou plusieurs capacités permettant non seulement à ce sujet d'invoquer l'opération, mais également aux autres sujets qui pourront être amenés à participer à la réalisation de cette opération en effectuant des opérations plus élémentaires. Il n'est donc pas nécessaire que ces derniers sujets possèdent des droits symboliques permanents dans la matrice pour ces autres opérations. Ceci limite le nombre des accès à la matrice globale. Dans cette vision, la notion de capacité est étroitement liée à la notion de permission et on voit que, pour les opérations complexes, l'existence d'une permission explicite mentionnée dans la matrice permet d'en déduire d'autres. Nous représentons ce fonctionnement par (26). Dans cette équation, on note qu'il est nécessaire d'établir une relation entre les types d'opérations et les différentes actions plus élémentaires qu'elles peuvent éventuellement impliquer. En revanche, le fonctionnement du système (c'est-à-dire la réalisation d'une action) est régi par l'existence d'une permission explicite et non d'un droit d'accès. Ceci explique la nécessité d'introduire la notion de capacité, et de transmission de ces capacités.

$$\begin{cases} s \times t \times o \Rightarrow \mathbf{P}(s \times t \times o) \\ \mathbf{P}(s \times t \times o) \Leftarrow s \times o \times a_t \\ \mathbf{P}(s' \times t' \times o') \Leftarrow \mathbf{P}(s \times t \times o) \wedge t \times t' \times s' \times o' \end{cases} \tag{26}$$

2.3.2.3 Un exemple de système informatique : UNIX

UNIX est un système d'exploitation communément utilisé dans les réseaux de stations de travail de moyenne ou grande taille. Ce système ne possède pas de fonctions particulières dédiées à la mise en œuvre d'une politique de sécurité assurant des propriétés fortes. Notamment, il est basé sur une politique d'autorisation discrétionnaire. Néanmoins ce système, multi-tâches et multi-utilisateurs, effectue un certain nombre de contrôles d'accès. Par ailleurs, la multitude de mécanismes de transfert de droits d'accès qu'il met à la disposition des utilisateurs, ainsi que les nombreuses vulnérabilités qui ont pu être identifiées dans ce système en font un cas d'étude particulièrement intéressant. En effet, il est fréquent d'identifier des situations contredisant les objectifs de sécurité du système. Ces problèmes peuvent avoir pour origine les principes du système d'exploitation, mais également les choix de configuration effectués par les utilisateurs. Dans tous les cas, la politique de sécurité doit être en mesure de représenter ces différents aspects, et éventuellement d'indiquer certaines actions correctrices. Notamment, l'intérêt d'une méthode d'évaluation de la sécurité permettant d'identifier et de justifier les actions correctrices à mettre en œuvre, qui fait l'objet du chapitre 3, apparaît alors clairement. Assurer la sécurité d'un système informatique basé sur UNIX impliquerait également l'ajout de mécanismes supplémentaires, en particulier des listes de contrôle d'accès.

2.3.2.3.1 Formalisation de certaines règles de fonctionnement d'UNIX

Les entités actives effectuant des opérations dans un système UNIX sont appelées des **processus**. Les entités contenant de l'information persistante sont constituées par des **fichiers**. Les processus peuvent manipuler ces fichiers, soit en y lisant ou écrivant des informations, soit en exécutant un fichier contenant le code d'un programme. Les processus sont identifiés par des numéros d'exécution, ou PID. Les fichiers sont caractérisés par des noms (contenant leur chemin d'accès). Un processus s'exécute généralement pour le compte d'un utilisateur, il est donc également caractérisé par le numéro d'identification d'un utilisateur, appelé UID. C'est ce numéro qui est utilisé dans les contrôles d'accès. Afin de simplifier la formulation des règles de fonctionnement d'UNIX, nous considérons l'ensemble des processus possédant le même UID comme un tout. Une entité active ou plusieurs entités actives concurrentes dans UNIX seront donc représentées par un même UID. On peut définir des groupes d'UID, qui seront également identifiés par des numéros, notés GID. À chaque fichier est également associé un UID, qui correspond au numéro d'identification du **propriétaire** du fichier, un GID qui correspond au groupe du fichier, et un ensemble de droits d'accès. Les principaux droits d'accès sont au nombre de neuf, et sont relatifs aux trois opérations de lecture, d'écriture et d'exécution pour chaque catégorie d'UID correspondant au propriétaire, au groupe, et à l'ensemble de tous les autres utilisateurs. Nous représenterons donc les règles de fonctionnement d'UNIX à partir de ces éléments, en considérant que les UID représentent les sujets dans UNIX, que les fichiers représentent les objets, et que nous avons neuf droits d'accès principaux. Les utilisateurs réels du système sont associés à chaque UID dans le système par une dernière relation indiquant qu'un utilisateur peut contrôler des processus correspondant à un certain UID. L'état initial de cette relation ne représente pas tout ce qui peut se passer dans le système, puisqu'il existe de nombreux mécanismes, licites ou illicites, permettant à un utilisateur de faire exécuter un processus possédant un autre UID. La figure 20 présente ces différents éléments dans la notation graphique présentée au 2.2.3.3.

Les règles de fonctionnement d'UNIX sont relativement simples. Les contrôles discrétionnaires d'un processus désirant accéder à un fichier sont effectués suivant les droits positionnés pour le fichier et les UID et GID respectifs du processus et du fichier. L'équation (27) présente les différentes règles de fonctionnement correspondant aux différentes opérations.

$$
\begin{aligned}
&\text{UID} \times \text{lecture} \times \text{Fichiers} \Rightarrow (\text{UID} \times \text{Fichiers} \wedge \text{r\_u} \times \text{Fichiers}) \\
&\quad \vee (\neg(\text{UID} \times \text{Fichiers}) \wedge \text{UID} \times \text{GID} \wedge \text{GID} \times \text{Fichiers} \wedge \text{r\_g} \times \text{Fichiers}) \\
&\quad \vee (\neg(\text{UID} \times \text{Fichiers}) \wedge \neg(\text{UID} \times \text{GID} \wedge \text{GID} \times \text{Fichiers}) \wedge \text{r\_o} \times \text{Fichiers}) \\
&\text{UID} \times \text{écriture} \times \text{Fichiers} \Rightarrow (\text{UID} \times \text{Fichiers} \wedge \text{w\_u} \times \text{Fichiers}) \\
&\quad \vee (\neg(\text{UID} \times \text{Fichiers}) \wedge \text{UID} \times \text{GID} \wedge \text{GID} \times \text{Fichiers} \wedge \text{w\_g} \times \text{Fichiers}) \\
&\quad \vee (\neg(\text{UID} \times \text{Fichiers}) \wedge \neg(\text{UID} \times \text{GID} \wedge \text{GID} \times \text{Fichiers}) \wedge \text{w\_o} \times \text{Fichiers}) \\
&\text{UID} \times \text{exécution} \times \text{Fichiers} \Rightarrow (\text{UID} \times \text{Fichiers} \wedge \text{x\_u} \times \text{Fichiers}) \\
&\quad \vee (\neg(\text{UID} \times \text{Fichiers}) \wedge \text{UID} \times \text{GID} \wedge \text{GID} \times \text{Fichiers} \wedge \text{x\_g} \times \text{Fichiers}) \\
&\quad \vee (\neg(\text{UID} \times \text{Fichiers}) \wedge \neg(\text{UID} \times \text{GID} \wedge \text{GID} \times \text{Fichiers}) \wedge \text{x\_o} \times \text{Fichiers})
\end{aligned}
\tag{27}
$$

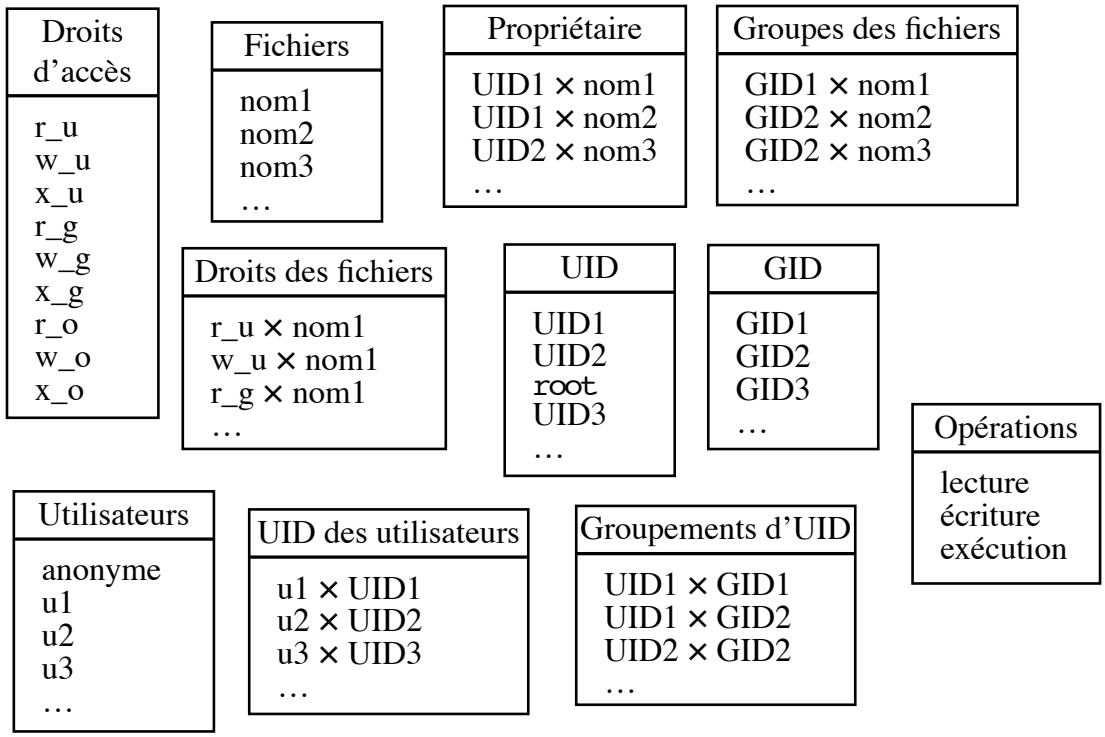


Figure 20 - Eléments de description pour UNIX

Dans UNIX, il existe également un utilisateur spécial, le super-utilisateur, qui n'est pas soumis aux contrôles d'accès. Pour les opérations de lecture, par exemple, on ajoute donc une règle qui signifie qu'un processus possédant l'UID appelé `root`[1] peut accéder en lecture à tous les fichiers. Il en est de même pour les autres types d'opération (28).

$$\begin{cases} \text{root} \times \text{lecture} \times \text{Fichiers} \\ \text{root} \times \text{écriture} \times \text{Fichiers} \\ \text{root} \times \text{exécution} \times \text{Fichiers} \end{cases} \tag{28}$$

Enfin, la politique d'autorisation utilisée dans UNIX étant une politique discrétionnaire, le propriétaire d'un fichier peut modifier les droits d'accès à ce fichier, ainsi que le groupe de ce fichier. Ceci est représenté par (29), où GID′ représente un ensemble copie de l'ensemble GID de la figure 20 (cf page 71).

$$\begin{aligned} &\text{UID} \times \text{Fichiers} \wedge \text{GID} \times \text{Fichiers} \Rightarrow \mathbf{P}(\text{Droits d'accès} \times \text{Fichiers}) \\ &\wedge \mathbf{P}(\neg\text{Droits d'accès} \times \text{Fichiers}) \wedge \mathbf{P}(\text{GID}' \times \text{Fichiers} \wedge \neg(\text{GID} \times \text{Fichiers})) \end{aligned} \tag{29}$$

1. Cet UID étant la valeur 0 dans UNIX.

2.3.2.3.2 Objectifs de sécurité

Dans le cadre d'UNIX, on souhaite avant tout contrôler que les différents utilisateurs du système ne puissent pas exécuter librement un processus dont l'UID ne reflète pas leur identification légitime, mais correspond à d'autres utilisateurs. Par exemple, on souhaite interdire aux utilisateurs ayant un accès anonyme au système de prendre le contrôle d'un processus possédant l'UID `root`, qui correspond au super-utilisateur. Un individu contrôlant un processus possédant l'UID `root` est en mesure d'effectuer des opérations exceptionnelles, comme par exemple de changer l'UID associé à un fichier. Ceci est représenté par (30), où UID' et UID'' représentent deux copies distinctes de l'ensemble UID de la figure 20 (cf page 71).

$$\mathrm{UID} \times \mathrm{Fichiers} \wedge \mathrm{UID}' \times \texttt{root} \Rightarrow \mathbf{P}(\mathrm{UID}'' \times \mathrm{Fichiers} \wedge \neg(\mathrm{UID} \times \mathrm{Fichiers})) \quad (30)$$

Ceci est représenté par l'objectif de sécurité (31).

$$\mathbf{F}(\mathrm{anonyme} \times \texttt{root}) \quad (31)$$

De la même manière, on peut souhaiter que l'utilisateur anonyme ne puisse contrôler des processus correspondant normalement aux différents administrateurs du système informatique, afin de protéger les données de ceux qui sont en charge du bon fonctionnement de ce système. Ceci sera représenté par (32), où "administrateurs" désigne un groupe d'utilisateurs.

$$\mathbf{F}(\mathrm{anonyme} \times \mathrm{UID} \wedge \mathrm{administrateurs} \times \mathrm{UID}) \quad (32)$$

2.3.2.3.3 Vulnérabilités considérées

Il est pourtant possible en utilisant certains mécanismes de transfert d'identité que des utilisateurs exécutent des processus correspondant normalement à d'autres utilisateurs. Par exemple, plusieurs des commandes d'UNIX permettent à un utilisateur u d'UID n de créer un processus dont l'UID n' est celui d'un autre utilisateur u' à condition qu'il soit fait mention de cette autorisation dans un fichier de configuration particulier, nommé `.rhosts`. La règle de fonctionnement correspondante est donc (33), où $\mathrm{UID} \times \mathrm{EstDansRhosts} \times \mathrm{UID}'$ représente l'autorisation mentionnée dans le fichier `.rhosts` entre u et u'.

$$\mathrm{UID} \times \mathrm{EstDansRhosts} \times \mathrm{UID}' \wedge \mathrm{Utilisateurs} \times \mathrm{UID} \Rightarrow \mathbf{P}(\mathrm{Utilisateurs} \times \mathrm{UID}') \quad (33)$$

Ce transfert d'identité ne résulte pas d'une défaillance du système. En effet, il s'agit ici d'un mécanisme parfaitement licite qui permet d'ailleurs à l'utilisateur u' d'autoriser u (auquel il fait confiance) à effectuer des opérations à sa place, sans pour autant devoir lui communiquer son mot de passe. En ce sens, ce mécanisme est donc un mécanisme destiné à améliorer la sécurité en évitant qu'un utilisateur se fasse totalement passer pour un autre du point de vue du système.

En revanche, dans les cas où le fichier `.rhosts` lui même peut être modifié par un autre utilisateur que son propriétaire, on peut considérer qu'il s'agit réellement d'une faille du système, due à une erreur de configuration. En effet, un utilisateur

pourra alors emprunter l'identité d'un autre dans le système en modifiant d'abord le fichier, puis en utilisant le mécanisme décrit par (33). Mais il n'y a aucune garantie que cette modification soit légitime. Cette possibilité est décrite par (34).

$$\text{UID} \times \text{écriture} \times .\text{rhosts} \wedge \text{UID}' \times .\text{rhosts} \Rightarrow \mathbf{P}(\text{UID} \times \text{EstDansRhosts} \times \text{UID}') \quad (34)$$

On peut également dire que, sous UNIX, quand un processus d'un utilisateur u' exécute un programme dont le code est contenu dans un fichier qui est modifiable par un processus appartenant à un autre utilisateur u, u est en mesure de prendre le contrôle du processus de u'. Ceci est représenté par (35). Ce type de mécanisme d'obtention de privilèges reflète le principe des chevaux de Troie (cf 1.1.1.2, page 8).

$$\begin{aligned} &\text{UID} \times \text{écriture} \times \text{Fichiers} \wedge \text{UID}' \times \text{exécution} \times \text{Fichiers} \\ &\wedge \text{Utilisateurs} \times \text{UID} \Rightarrow \mathbf{P}(\text{Utilisateurs} \times \text{UID}') \end{aligned} \quad (35)$$

Un problème délicat des transferts de privilèges réside dans leur transitivité. En effet, il est possible qu'un utilisateur u, qui ne peut normalement pas obtenir directement des privilèges de l'utilisateur u'', puisse obtenir ceux d'un autre utilisateur u' lequel peut accéder à des moyens de dérober des privilèges de u''. C'est ce problème qui est à l'origine de la représentation des vulnérabilités du système sous forme de graphe des privilèges, que nous avons vu précédemment (cf 1.3.3.3, page 50). Dans le cas d'UNIX, de nombreuses vulnérabilités existent [Garfinkel & Spafford 1996], et leur intégration dans la politique de sécurité a toutes les chances de montrer que les objectifs de sécurité ne sont pas garantis. De plus, ainsi que nous l'avons vu, certains mécanismes de transfert de privilèges constituent des fonctionnalités commodes et utiles pour le fonctionnement du système; leur élimination peut donc créer des difficultés inacceptables pour les utilisateurs. Face à ceci, il importe d'être en mesure d'identifier les vulnérabilités qui représentent une menace importante pour la sécurité, afin de pouvoir se concentrer sur leur élimination. Une méthode d'évaluation de la sécurité prenant en compte ces vulnérabilités est alors nécessaire.

2.4 Perspectives

L'utilisation de la logique modale pour la spécification d'une politique de sécurité que nous avons présentée n'exploite pas toutes les opportunités offertes par les différentes théories modales. En fait, l'approche que nous présentons laisse ouvertes un certain nombre de perspectives d'extension:

- Tout d'abord, l'intérêt pratique d'utiliser les opérateurs modaux **O**, **P** et **F** de la logique déontique pour formuler naturellement et rigoureusement les objectifs de la politique de sécurité laisse supposer que l'utilisation d'autres opérateurs modaux pourrait s'avérer très utile pour la construction de la politique de sécurité d'un système. Ainsi que nous l'avons mentionné au 2.2.2, et ainsi que l'on peut le voir au travers des travaux rapportés dans l'annexe A (page 153), la notion de modalité peut être étendue au delà des seuls points de vue ontique et

déontique. L'introduction des modalités dynamiques, temporelles, épistémiques ou doxastiques, au-delà de leurs désignations ésotériques, correspond en effet respectivement à l'introduction dans le langage logique des notions d'action ou d'agent, de temps, de connaissance et de croyance. L'introduction simultanée de plusieurs opérateurs modaux de nature différente permet d'obtenir un langage extrêmement expressif capable de transcrire précisément des énoncés complexes comme : "*a* croit qu'il est interdit que *b* connaisse *p*". Dans l'optique de l'étude de la sécurité, nous nous sommes concentrés sur l'étude des opérateurs déontiques car la notion d'obligation est la notion centrale. Toutefois, l'utilisation simultanée d'autres opérateurs modaux est extrêmement tentante tout au long de la formulation d'une spécification. Mais ceci correspondrait à étendre le langage monomodal que nous avons présenté et à utiliser une logique multimodale. Certains résultats encourageants présentés dans l'annexe A incitent à franchir ce pas, ce qui ouvrirait de nombreux axes d'études. Par exemple, du point de vue de la sécurité :

- L'utilisation simultanée d'opérateurs ontiques ($\Box$, $\Diamond$) et déontiques (**O**, **P**, **F**) rendrait explicite la distinction que l'on peut faire entre un objectif ou une règle de sécurité du système et une vulnérabilité. En effet, la première notion correspond à ce qui est obligatoire et la deuxième à ce qui est possible.
- L'introduction de la notion de temps dans une politique de sécurité semble incontournable dès que l'on s'intéresse aux aspects dynamiques des systèmes d'information. La logique du premier ordre, ou une logique d'intervalle, permet d'aborder ce problème dans le cadre monomodal, mais on peut également envisager d'utiliser la logique temporelle (opérateurs **F**, **G**, **P**, **H** [Van Benthem 1983]) pour exprimer des contraintes plus fortes [Yu & Gligor 1988 ; Maibaum 1993].
- L'étude de la logique déontique dans la littérature a montré l'intérêt d'utiliser une logique de l'action combinant des opérateurs déontiques et dynamiques pour représenter les obligations de chaque agent présent dans le système d'information [Hilpinen 1993 ; Jones & Sergot 1993]. La logique de l'action trouve naturellement sa place dans un cadre multimodal (opérateurs $[a]$, $\langle a \rangle$ où $a \in \{A_1, A_2, \ldots\}$ désigne un agent).
- Enfin, du point de vue théorique, les logiques épistémique (opérateurs **K**, **C**) et doxastique (opérateur **B**) devraient certainement être associées au cadre formel utilisé pour définir une politique de sécurité. En effet, on comprend que la connaissance des obligations (ou des interdictions) par les individus concernés par la politique de sécurité est un point délicat pour sa mise en œuvre. Par ailleurs, ces opérateurs peuvent être utilisés pour représenter plus précisément des propriétés de confidentialité [Bieber & Cuppens 1993 ; Cuppens 1993b].

- Plus généralement, l'application satisfaisante d'une des nombreuses logiques modales à une tâche de spécification particulière telle que celle des besoins de sécurité d'un système d'information, tout comme l'utilisation de la logique modale temporelle dans d'autres domaines [Emerson 1990], incite à considérer les théories modales comme des outils prometteurs pour la représentation rigoureuse de propriétés complexes. Face à ces résultats, et en considérant que la logique multimodale est un cadre unificateur susceptible de préserver les atouts des nombreuses théories modales déjà étudiées [Catach 1989], l'étude de son utilisation pour la spécification des systèmes, notamment pour ce qui est de leurs propriétés non fonctionnelles, semble tout à fait digne d'intérêt.

Chapitre 3 Évaluation de la sécurité

Même si elle permet de définir précisément les règles et les propriétés de sécurité attendues dans un système d'information, la politique de sécurité définit avant tout un fonctionnement idéal. Le fonctionnement réel du système est soumis par ailleurs à des contraintes de faisabilité ou de souplesse, et peut s'avérer en contradiction avec la politique de sécurité. L'observation du système en exploitation peut donc révéler des éléments nouveaux susceptibles de mettre en défaut les objectifs de sécurité. Ces différents éléments constituent alors des *vulnérabilités* du système. Dans la pratique, en raison des nécessités du fonctionnement de l'organisation ou en regard des efforts nécessaires à leur élimination, ces conflits peuvent même être acceptables. L'élimination de toutes les vulnérabilités peut donc poser des difficultés, notamment dans le cas où elles trouvent leur origine dans des fonctionnalités contribuant à l'efficacité du système d'information. Enfin, on conçoit que, dans certains cas, l'impact de ces vulnérabilités sur la sécurité effective du système soit négligeable, ce qui justifie qu'elles subsistent dans l'organisation. Toutefois, afin d'estimer précisément l'impact de vulnérabilités qui entrent en conflit avec la politique de sécurité préalablement définie, il importe de disposer d'une méthode d'évaluation de la sécurité. Cette évaluation de la sécurité offre alors un moyen objectif de justifier la confiance que l'on accorde au système malgré la présence de vulnérabilités.

3.1 Une méthode d'évaluation quantitative

Nous présentons dans cette section une méthode d'évaluation quantitative de la sécurité appuyée sur une représentation des vulnérabilités du système sous forme de *graphe des privilèges* (cf 1.3.3.3, page 50). Tout d'abord, nous décrivons les principales étapes correspondant à la mise en œuvre de cette méthode. Ensuite, nous montrons au travers d'un exemple simple comment le modèle du graphe des privilèges peut être utilisé, en conjonction avec une spécification formelle de la politique de sécurité dans le langage de la logique modale, pour représenter les vulnérabilités du système. Enfin, nous étudions les différentes mesures quantitatives qui peuvent être définies à partir du graphe des privilèges et les hypothèses correspondantes.

3.1.1 Présentation de la méthode

3.1.1.1 Intégration de vulnérabilités dans la description de la sécurité

La politique de sécurité du système d'information s'attache avant tout à préciser son comportement *attendu*. Afin d'obtenir une évaluation de la sécurité, on doit également prendre en compte les vulnérabilités résiduelles pouvant exister dans le système. Ces vulnérabilités peuvent avoir pour origine certaines incohérences de la politique de sécurité (cf 2.1.4, page 61) ou être reliées à la force des mécanismes de sécurité utilisés dans le système[1]. Dans le deuxième cas, on peut ainsi considérer, par exemple, qu'un mécanisme d'authentification par mot de passe est intrinsèquement vulnérable, et possède une force qui dépend des mots de passe choisis (et notamment de leur taille) ainsi que des algorithmes cryptographiques utilisés pendant la procédure d'authentification. Une méthodologie d'évaluation de la force de ces mécanismes est présentée dans [ITSEM 1993, §6.C.28-34]. Dans le premier cas, les incohérences entre la politique de sécurité (qui décrit un fonctionnement idéal) et le fonctionnement réel du système peuvent donner lieu à des situations où les propriétés de sécurité attendues ne sont pas obtenues. De telles faiblesses peuvent apparaître dans le système au moment de sa construction, ou pendant sa vie opérationnelle. Notamment dans le cas où la spécification de la politique de sécurité est effectuée vis-à-vis d'un système existant, ces vulnérabilités sont généralement constatées a posteriori en observant le système ou en analysant ses défaillances. Dans tous les cas, une évaluation de la sécurité (qui diffère de la validation de la politique de sécurité ou du système d'information) n'a de sens qu'en présence de vulnérabilités dans le système. La spécification de la politique de sécurité, qui définit précisément les règles et les propriétés qui devraient être respectées par le système, doit donc être complétée par une représentation des vulnérabilités, qui tient compte de l'état réel du système et notamment de sa configuration et des fautes qui ont été identifiées.

Ainsi que nous l'avons vu au chapitre 2, la politique de sécurité permet d'identifier les différentes permissions qui sont définies dans le système ainsi que les conditions sous lesquelles ces permissions peuvent être légitimement accordées. De manière analogue à une règle de sécurité, une vulnérabilité correspond à la possibilité de réaliser une action moyennant un certain nombre de conditions sur l'état du système. Néanmoins, dans le cadre de l'exploitation d'une vulnérabilité, les conditions de réalisation de cette action diffèrent de celles spécifiées par les règles de sécurité. Notamment, ces conditions peuvent ne pas dépendre de l'état de sécurité du système mais seulement de certaines caractéristiques du système sans rapport avec la sécurité (comme la version d'un logiciel, ou l'absence d'un supérieur par exemple). Dans ce cas, une vulnérabilité représente la possibilité pour un ou plusieurs utilisa-

1. Ceci correspond à la distinction effectuée dans l'ITSEM entre les *mécanismes de type B* et *de type A* respectivement (cf 1.3.1.2, page 39, et [ITSEM 1993, §6.C.4-6.C.7]).

teurs du système d'outrepasser les autorisations légitimes telles qu'elles sont définies par les règles de sécurité. Une vulnérabilité peut également correspondre à une règle de sécurité qui permet dans le cadre d'une utilisation légitime, de mettre en œuvre une fonction du système (comme par exemple une délégation de pouvoirs). Ces différentes règles de sécurité peuvent éventuellement être utilisées de manière à amener le système dans un état qui enfreint un des objectifs de sécurité. Dans ce cas, c'est un des mécanismes même du système, par ailleurs accepté, qui peut constituer une vulnérabilité.

Qu'elles soient acceptées parmi les fonctions du système, ou qu'elles aient été constatées a posteriori, l'important est alors de prendre en compte ces vulnérabilités et de déterminer à quel point il est probable qu'une utilisation illégitime de ces mécanismes conduise à mettre en défaut les objectifs de sécurité.

3.1.1.2 Détermination des vulnérabilités à prendre en compte

L'identification des vulnérabilités du système que l'on souhaite prendre en compte dans le cadre de l'évaluation de sa sécurité doit s'appuyer sur un certain nombre d'informations.

La politique de sécurité du système définit déjà, dans les règles de sécurité, des mécanismes permettant d'accorder des permissions et donc de réaliser les actions associées. Ces mécanismes peuvent éventuellement constituer des vulnérabilités s'ils permettent d'enfreindre les objectifs de sécurité. La définition de la politique de sécurité permet donc d'ores et déjà d'identifier un certain nombre des vulnérabilités.

Toutefois, d'autres sources d'informations peuvent permettre de compléter celles fournie par la politique de sécurité et notamment de prendre en compte le fonctionnement.

D'une part, on peut disposer de la liste d'un certain nombre de vulnérabilités déjà identifiées dans des systèmes de même nature que le système cible de l'évaluation. Par exemple, dans le cas d'un système informatique comme UNIX, on dispose de références relatives à un certain nombre de problèmes pratiques qui peuvent compromettre la sécurité [Aslam 1995; Garfinkel & Spafford 1996]. Dans le cas d'une organisation, l'étude de la sécurité ou des éventuelles défaillances que d'autres organisations similaires ont pu connaître fournit une identification des vulnérabilités potentiellement existantes dans l'organisation. Enfin, certaines méthodes génériques permettant de mettre en défaut les mécanismes de sécurité, comme le principe d'un cheval de Troie ou d'un canal caché, fournissent des éléments qui permettent de rechercher précisément dans le système certaines vulnérabilités.

D'autre part, dans le cadre d'une organisation, la définition de la politique de sécurité passe par une description générale du fonctionnement de l'organisation. Cette description de haut niveau permet d'identifier les différentes étapes de la réalisation d'une action. Afin d'identifier des vulnérabilités éventuelles, on peut également s'intéresser à ces différentes étapes et envisager s'il est possible de court-circuiter

certaines opérations et de tirer parti du fonctionnement normal pour faire en sorte qu'une action soit réalisée sans que les contrôles de sécurité habituels aient lieu. À chaque étape, on peut juger de la difficulté d'initier ce fonctionnement dans une perspective malveillante en l'absence de toute opération préalable et éventuellement décider de l'existence d'une vulnérabilité. Comme au 2.3.1.1 (page 83), cette analyse peut partir d'une description des différents flux d'information qui existent dans l'organisation.

Enfin, l'analyse des objectifs de sécurité définis pour l'organisation ou le système étudié permet d'entamer une recherche des vulnérabilités présentes dans le système à partir des éléments ou des opérations que l'on souhaite plus particulièrement protéger. En effet, dans le cas d'un système d'information au fonctionnement complexe, il peut être plus profitable d'étudier directement les objectifs de sécurité et d'envisager des moyens de les mettre en défaut que de s'attacher à analyser l'ensemble des flux d'information. Dans ce cas, on fait le choix de concentrer l'analyse sur les propriétés attendues du système. La recherche des vulnérabilités présentes dans le système s'apparente alors à une procédure de validation telle qu'elle peut être mise en œuvre dans le cadre d'une évaluation de la sécurité selon des critères normalisés (cf 1.3.1, page 38).

Une fois que les différents types de vulnérabilités qui doivent être pris en compte pour l'évaluation de la sécurité du système ont été identifiés, il s'agit de vérifier directement dans le système s'il est effectivement possible de les mettre en œuvre. Ceci conduit à définir précisément les conditions qui doivent être remplies pour que ces vulnérabilités soit exploitables, et dans ce cas, quels sont les utilisateurs du système qui seront en mesure de les exploiter et à quelle fin.

3.1.1.3 Quantification des vulnérabilités

Dans une première étape, des valeurs quantitatives sont associées aux vulnérabilités élémentaires. Par exemple, on affecte à chaque arc du graphe des privilèges (cf 1.3.3.3, page 50) un poids correspondant à "l'effort" nécessaire à un attaquant potentiel pour exploiter la méthode de transfert de privilèges correspondant à cet arc. Cette notion d'effort regroupe les différentes caractéristiques du processus d'attaque, comme l'existence d'outils publics préexistants, le temps nécessaire pour mettre en œuvre l'attaque, la puissance de calcul disponible pour l'attaquant, etc. [ITSEM 1993, §6.C.30 ; Littlewood *et al*. 1993]. Par exemple, l'effort nécessaire pour obtenir le mot de passe d'un utilisateur peut être évalué en fonction de la puissance de calcul et du temps nécessaire à un programme comme `crack` [Muffet 1992] pour deviner ce mot de passe. Pour une attaque par utilisation d'un cheval de Troie, l'effort peut être évalué en fonction de la compétence nécessaire pour concevoir le cheval de Troie, le temps nécessaire pour l'implanter dans un programme exécuté par l'utilisateur cible de l'attaque, et le temps moyen nécessaire pour que l'utilisateur active ce programme (ce dernier paramètre ne dépendant pas du processus

d'attaque mais seulement du comportement de l'utilisateur). La valeur d'effort attribuée à un arc est donc un paramètre complexe, qui est représenté par le taux de succès de l'attaque élémentaire correspondante.

Idéalement, le choix d'un ensemble de variables significatives devrait se baser sur l'analyse d'intrusions connues et des paramètres qui ont influencé leur déroulement. Malheureusement, de telles données sont très difficiles à obtenir précisément. Par ailleurs, il semble difficile de faire abstraction du profil de l'attaquant exploitant une vulnérabilité particulière pour obtenir une évaluation quantitative significative. Toutefois, on peut envisager de rassembler les expertises relatives aux différentes vulnérabilités constatées dans les systèmes, qu'il s'agisse de systèmes informatiques ou de systèmes d'information, et d'utiliser un certain nombre de paramètres généraux afin d'obtenir une valeur quantitative représentative de chaque vulnérabilité considérée. Ces différents aspects sont détaillés plus précisément dans [Dacier 1994, §IV.1].

L'ITSEM fournit également un cadre d'analyse bien délimité pour l'évaluation quantitative de ces vulnérabilités élémentaires. En effet, ainsi que nous l'avons déjà mentionné (cf 1.3.1.2, page 39) l'ITSEM précise la manière dont on peut estimer le niveau de résistance des mécanismes de sécurité [ITSEM 1993, §3.3.29-32, §6.C.28-34]. Pour ce qui est des critères, cette évaluation reste ordinale. Néanmoins, la justification de la confiance accordée à la résistance des mécanismes de protection telle qu'elle doit être effectuée dans le cadre d'une évaluation pour les ITSEC met en évidence les différents paramètres qui peuvent être pris en compte. Notamment, l'ITSEM identifie trois dimensions principales permettant de juger de la résistance des mécanismes de sécurité: la compétence, les ressources (parmi lesquelles on retrouve le temps, ou l'équipement nécessaire) et les opportunités (collusion, chance, possibilité de détection). Une analyse effectuée dans le cadre des ITSEC contient donc une étude critique complète des mécanismes de sécurité utilisés dans le système cible de l'évaluation. On peut se baser sur cette étude afin de proposer une quantification des autres vulnérabilités. Des paramètres analogues peuvent en effet être utilisés afin de qualifier l'effort nécessaire pour exploiter une vulnérabilité issue d'une incohérence entre la politique de sécurité et le fonctionnement réel du système. À l'évaluation ordinale selon les trois niveaux de résistance: élémentaire, moyenne et élevée proposée par les ITSEC [ITSEC 1991, §3.6-3.8] se substitue alors une quantification prenant en compte les différentes dimensions d'analyse proposées dans l'ITSEM. Les modalités exactes de quantification des vulnérabilités peuvent également être adaptées au contexte de l'organisation considérée, comme on peut le voir dans [Aerospatiale 1997].

Dans la perspective d'une évaluation de la sécurité opérationnelle destinée à surveiller l'*évolution* de la sécurité et à identifier les failles de sécurité importantes qui peuvent apparaître dans le système, il est plus simple, dans une première étape, de regrouper les vulnérabilités considérées dans différentes classes correspondant à leur facilité d'exploitation. La valeur numérique attribuée à chaque classe correspond alors à une classification qualitative des vulnérabilités. Les valeurs des diffé-

rentes classes diffèrent par exemple d'un ordre de grandeur. Ceci permet d'exploiter les résultats numériques de l'évaluation afin d'observer l'évolution de la sécurité, et garantit que les vulnérabilités considérées comme les plus dangereuses seront celles qui auront le plus d'influence sur la mesure calculée. Étant donné qu'une valeur particulière peut en fait être attribuée individuellement à chaque vulnérabilité, le nombre et la finesse des classes considérées ne sont pas fixés a priori par le modèle. Il est donc possible de raffiner progressivement la classification utilisée en fonction de l'expérience acquise dans l'étude de l'évolution opérationnelle de la sécurité du système, de la disponibilité d'informations extérieures, ou encore de la possibilité d'utiliser des outils d'analyse du système permettant d'estimer précisément les paramètres caractéristiques de chacune des vulnérabilités rencontrées.

3.1.1.4 Évaluation quantitative

Une évaluation quantitative de la sécurité, appuyée sur la politique de sécurité et sur une représentation des vulnérabilités présentes dans le système, demande donc de disposer d'éléments de quantification élémentaires. À partir de ces valeurs élémentaires, l'évaluation quantitative consiste à obtenir une valeur globale tenant compte de l'ensemble des différentes vulnérabilités présentes dans le système. Cette valeur doit être représentative de la difficulté de mettre en défaut les objectifs de sécurité du système à l'aide de ces vulnérabilités pour l'ensemble des attaquants possibles. Le résultat de cette évaluation dépend donc des valeurs quantitatives attribuées aux différentes vulnérabilités mais aussi de la manière dont elles peuvent être combinées pour mettre en défaut les objectifs de la politique de sécurité. En effet, dans la plupart des cas, il est probable que la mise en œuvre d'une seule vulnérabilité ne permette pas directement de mettre en défaut un objectif de sécurité du système mais constitue seulement une étape dans cette direction. Il est donc alors nécessaire, afin d'évaluer la sécurité du système, de tenir compte de l'ensemble des vulnérabilités qu'il contient et de la possibilité d'exploiter des combinaisons de plusieurs d'entre elles.

3.1.2 Représentation des vulnérabilités : Graphe des privilèges

Afin d'obtenir une évaluation quantitative de la sécurité, nous allons utiliser le graphe des privilèges comme modèle de représentation des vulnérabilités d'un système [Dacier 1994 ; Dacier & Deswarte 1994]. Le graphe des privilèges est donc un modèle du système en exploitation, intégrant ses vulnérabilités, alors que la politique de sécurité représente le fonctionnement idéal de ce système. Nous présentons dans cette section les connections que l'on peut effectuer entre la spécification de la politique de sécurité du système effectuée dans le langage de la logique déontique et la représentation des vulnérabilités sous forme de graphe des privilèges.

3.1.2.1 Définition d'une vulnérabilité

Dans le graphe des privilèges, une vulnérabilité correspond à une méthode de transfert de privilèges. Les nœuds du graphe représentent les différents privilèges qui existent dans le système. Un arc est donc créé entre deux nœuds si une méthode permet effectivement, possédant les privilèges du nœud origine d'obtenir ceux du nœud destination. L'existence d'un arc entre deux nœuds dépend donc de l'état du système à un instant donné qui peut ou non permettre l'exploitation d'une certaine vulnérabilité.

Dans le langage logique de spécification que nous avons proposé, cette représentation correspond aux définitions des notions de privilèges et de vulnérabilités données par les équations (12) et (13) (cf 2.2.4.4, page 78). Étant donné un ensemble de permissions P_w vraies dans un certain monde *w*, une vulnérabilité correspond donc à l'existence d'un monde possible w' en relation avec *w* contenant un autre ensemble de permissions $P'_{w'}$, moyennant une condition *p* portant sur l'état du système dans *w*.

3.1.2.2 Construction directe du graphe

Quand c'est possible, la représentation des vulnérabilités du système en exploitation peut s'appuyer directement sur le modèle du graphe des privilèges. Dans ce cas, il est nécessaire de pouvoir identifier au préalable dans le système les différents privilèges à partir desquels peuvent être définies les vulnérabilités que l'on souhaite prendre en compte. La construction du graphe des privilèges s'effectue alors par une analyse directe du système consistant à tester les conditions d'existence d'un arc entre deux nœuds du graphe, c'est-à-dire d'une méthode de transfert de privilèges permettant d'obtenir, à partir des privilèges correspondant au nœud origine, les privilèges du nœud destination.

Les objectifs de sécurité définis par la politique de sécurité permettent enfin d'identifier les différents nœuds du graphe correspondant aux attaquants et aux cibles potentiels qui sont pertinents à étudier dans un système donné. Quand il est aisé d'identifier tous les privilèges existant dans le système, comme c'est notamment le cas dans un système informatique, cette construction directe d'un graphe des privilèges représentant les vulnérabilités du système est alors la solution la plus commode permettant d'aboutir à une évaluation quantitative de la sécurité.

3.1.2.3 Exploitation de la spécification logique

Si on représente à la fois la politique de sécurité et les vulnérabilités du système dans le langage de la logique déontique, il est possible de construire le modèle correspondant dans la sémantique de Kripke. Ce modèle constitue alors un modèle du système mentionnant ses différentes évolutions possibles du point de vue de la sécurité, compte tenu des mécanismes de sécurité et des vulnérabilités considérés. L'intégration de l'état courant du système à cette description permet également de limiter le modèle obtenu aux évolutions démarrant de cet état. Nous présentons

dans cette section une méthode simple de construction du modèle de Kripke associé à un ensemble de formules modales: la méthode des tableaux. Ainsi que nous le verrons au travers d'un exemple, l'utilisation de cette méthode de déduction permet alors d'identifier les différentes permissions existant dans les mondes du modèle construit, ainsi que la relation d'accessibilité reliant ces mondes. À partir de ces informations, on peut déterminer si, étant donné un ensemble de permissions, c'est-à-dire un ensemble de privilèges, il est possible d'atteindre une situation dans laquelle d'autres permissions sont vraies, c'est-à-dire si on peut obtenir de nouveaux privilèges.

Le modèle obtenu décrit toutes les différentes évolutions possibles du système, et notamment celles qui peuvent conduire à l'obtention de permissions enfreignant les objectifs de sécurité du système. Il fournit donc des informations analogues à celles qui peuvent être déduites du graphe des privilèges en construisant le processus d'intrusion (cf figure 6, page 53) permettant, à partir d'un nœud du graphe, d'atteindre un autre nœud.

3.1.2.3.1 Présentation de la méthode des tableaux

La méthode des tableaux sémantiques est une méthode de déduction utilisable dans le cadre d'une logique modale et qui fournit également à la suite de son application un modèle correspondant à l'ensemble de formules modales étudié. Le calcul des séquents de Gentzen et la méthode des tableaux peuvent être vus comme des variantes notationnelles l'un de l'autre. Plusieurs méthodes basées sur les tableaux sont applicables dans le cadre d'une logique modale. Nous présentons ici une version issue de [Fitting 1993, §1.9], qui définit des noms pour les différents mondes possibles du modèle. Ces noms sont appelés des **préfixes**. On note σX , où σ est un préfixe et X une formule, le fait que X soit vraie dans le monde identifié par σ. σX est appelée une formule préfixée. Dans un modèle $M = \langle W, R, V \rangle$, deux mondes possibles peuvent ou non être associés par la relation d'accessibilité R. Il est souhaitable de choisir un système de préfixes qui permettent d'indiquer syntaxiquement si deux mondes identifiés par leurs préfixes sont en relation. Un préfixe est donc simplement une suite finie non-vide d'entiers positifs, telle que $\langle 1, 3, 2, 1, 4 \rangle$. Le monde initial du modèle est alors noté $\langle 1 \rangle$. Si n est un entier positif et σ un préfixe, on note σn le préfixe résultant de l'adjonction de n à σ. Par exemple, si $\sigma = \langle 1, 3, 2, 1, 4 \rangle$, on a $\sigma 3 = \langle 1, 3, 2, 1, 4, 3 \rangle$. Enfin, un préfixe de la forme σn est *accessible* depuis le préfixe σ.

Une preuve par la méthode des tableaux prend la forme d'un arbre où chaque nœud est associé à une formule préfixée. Des règles régissent l'initialisation de l'arbre, sa progression, et l'arrêt de la construction. Tout d'abord, nous définissons la notion de satisfiabilité pour un tableau.

Un tableau est satisfiable si une de ses branches est satisfiable. Une branche est satisfiable si l'ensemble des formules préfixées figurant sur cette branche est satisfiable. Enfin, un ensemble S de formules préfixées est satisfiable s'il existe un modèle de Kripke $\mathrm{M} = \langle \mathrm{W}, \mathrm{R}, \mathrm{V} \rangle$ et si on peut construire une fonction N associant les préfixes apparaissant dans S aux mondes de W telle que :

1) pour tout σ, τ apparaissant dans S, si le préfixe τ est accessible depuis σ, alors $N(\sigma)\mathrm{R}N(\tau)$;

2) si $\sigma X \in S$ alors $\models^{\mathrm{M}}_{N(\sigma)} X$.

Informellement, un ensemble de formules préfixées est satisfiable s'il décrit partiellement un certain modèle.

Les tableaux sont des réfutations. Pour prouver X, on fait l'hypothèse que X est fausse, et on dérive une contradiction. Ceci permet de conclure que X doit être vraie. Une preuve de X commence avec un tableau trivial contenant seulement un nœud racine, associé à $\langle 1 \rangle \neg X$. Si X n'est pas vraie, l'ensemble $\{\langle 1 \rangle \neg X\}$ est satisfiable, et nous démarrons donc la construction avec un tableau satisfiable. Des règles permettent ensuite d'étendre l'arbre en étendant ses branches, et ces règles préservent la satisfiabilité. L'application exhaustive de toutes ces règles permet de contredire (ou de vérifier) la satisfiabilité de $\{\langle 1 \rangle \neg X\}$, et donc de conclure sur la validité de X.

On dit qu'une branche du tableau est fermée si elle contient à la fois σX et $\sigma \neg X$ pour une formule X quelconque (ou si elle contient $\sigma \bot$ ou $\sigma \neg \top$ avec le langage de l'annexe A). Un tableau est fermé si toutes ses branches sont fermées. Un tableau fermé commençant par $\langle 1 \rangle \neg X$ constitue une preuve de X.

Les règles d'extension d'un tableau concernent à la fois les opérateurs de la logique propositionnelle classique et les opérateurs modaux. Certains opérateurs conduisent à l'introduction de nouvelles branches dans l'arbre. Les différentes règles applicables sont présentées schématiquement dans le tableau 7.

3.1.2.3.2 Exemple d'application

À partir d'une spécification de la politique de sécurité du système, l'utilisation d'une méthode de déduction comme la méthode des tableaux permet de construire un modèle identifiant les différentes évolutions possibles du système du point de vue de la sécurité. Nous présentons à présent au travers d'un exemple les informations qui peuvent être obtenues de cette manière. Ainsi que nous allons le voir, ces informations sont analogues à celles que l'on peut obtenir à partir d'un graphe des privilèges représentant directement les vulnérabilités observées dans le système et en dérivant le processus d'intrusion correspondant à un attaquant et une cible particuliers.

Afin d'illustrer cette construction, nous reprenons un exemple présenté dans [Dacier 1994, §II.5.2] correspondant à un système comportant un faible nombre de sujets et d'objets et un schéma d'autorisation simple composé de deux règles. Les

$$\frac{\sigma \neg \neg X}{\sigma X}$$

$$\frac{\sigma(X \Rightarrow Y)}{\sigma \neg X \mid \sigma Y} \qquad \frac{\sigma \neg (X \Rightarrow Y)}{\begin{array}{c}\sigma X \\ \sigma \neg Y\end{array}}$$

$$\frac{\sigma(X \wedge Y)}{\begin{array}{c}\sigma X \\ \sigma Y\end{array}} \qquad \frac{\sigma \neg (X \wedge Y)}{\sigma \neg X \mid \sigma \neg Y}$$

$$\frac{\sigma(X \vee Y)}{\sigma X \mid \sigma Y} \qquad \frac{\sigma \neg (X \vee Y)}{\begin{array}{c}\sigma \neg X \\ \sigma \neg Y\end{array}}$$

$$\frac{\sigma \mathbf{O} X}{\sigma n X} \qquad \frac{\sigma \neg \mathbf{O} X}{\sigma n \neg X}$$

pour chaque σn existant — avec un nouveau σn

$$\frac{\sigma \mathbf{P} X}{\sigma n X} \qquad \frac{\sigma \neg \mathbf{P} X}{\sigma n \neg X}$$

avec un nouveau σn — pour chaque σn existant

Tableau 7 - Règles de la méthode des tableaux

éléments de description de la politique de sécurité sont représentés graphiquement dans la figure 21. Les règles de sécurité correspondant au schéma d'autorisation sont données par R1 et R2. L'état initial de protection dans le monde initial w_0 , qui correspond à l'état de sécurité du système représenté par une matrice de contrôle d'accès est également donné sous forme graphique dans la figure 21.

$$\text{sujet} \times \text{obj1} \times \text{o} \wedge \text{autre sujet} \times \text{obj1} \times \text{e} \wedge \text{autre sujet} \times \text{obj3} \times \text{r} \Rightarrow \mathbf{P}(\text{sujet} \times \text{obj3} \times \text{r}) \qquad \text{R1}$$

$$\begin{cases} \text{sujet} \times \text{obj2} \times \text{w} \wedge \text{autre sujet} \times \text{obj2} \times \text{o} \wedge \text{autre sujet} \times \text{obj3} \times \text{r} \Rightarrow \mathbf{P}(\text{sujet} \times \text{obj3} \times \text{r}) \\ \text{sujet} \times \text{obj2} \times \text{w} \wedge \text{autre sujet} \times \text{obj2} \times \text{o} \wedge \text{autre sujet} \times \text{obj3} \times \text{w} \Rightarrow \mathbf{P}(\text{sujet} \times \text{obj3} \times \text{w}) \end{cases} \qquad \text{R2}$$

Ces deux règles signifient respectivement que:

- si un utilisateur 'sujet' possède un objet *obj* de type 'obj1' et si un utilisateur 'autre sujet' peut exécuter cet objet *obj*, alors 'autre sujet' peut accorder à 'sujet' tous les droits en lecture que 'autre sujet' détient sur les objets de type 'obj3';
- et, si un utilisateur 'sujet' peut écrire dans un objet *obj* de type 'obj2' et si un utilisateur 'autre sujet' est le propriétaire de *obj*, alors 'autre sujet' peut accorder à 'sujet' tous les droits en lecture et écriture que 'autre sujet' détient sur les objets de type 'obj3'.

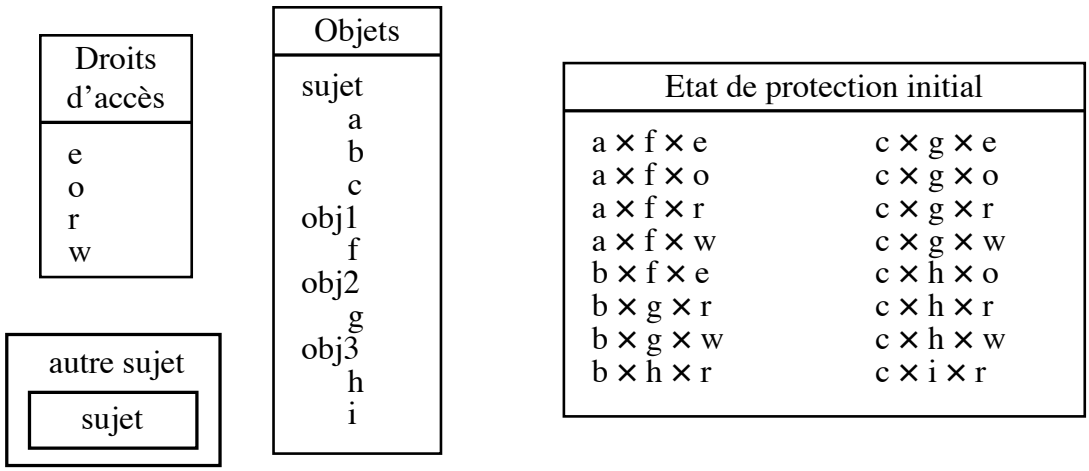


Figure 21 - Éléments de description et état de protection initial

Comme nous avons affaire à une politique de sécurité basée sur une matrice de contrôle d'accès, les triplets $s \times obj \times a_t$ où s représente un sujet, obj un objet et a_t un droit d'accès correspondent directement à des permissions. Les règles R1 et R2 permettent donc d'effectuer des transferts de permissions. Étant donné l'axiome d'interaction (15), ces règles conduisent à des règles de la forme (14) correspondant à une vulnérabilité.

Nous étudions deux objectifs de sécurité du système qui sont représentés par les règles P1 et P2.

$$\mathbf{F}(a \times i \times r) \qquad \text{P1}$$

$$\mathbf{F}(a \times h \times w) \qquad \text{P2}$$

Les règles R1 et R2, écrites en notation étendue, donnent lieu à la génération directe de 48 règles correspondant aux différents sujets et objets présents dans le système. Étant donné les informations dont nous disposons concernant l'état de protection initial dans le système, ce nombre peut être diminué en simplifiant ces règles, ce qui permet de diminuer la taille des informations à associer à la racine du tableau généré. Ainsi, si nous appliquons successivement les règles R1 et R2 à partir de l'état de protection initial (dans le monde w_0) nous pouvons effectuer les déductions suivantes :

$$\begin{array}{c}
\dfrac{\text{R1} \wedge (a \times f \times o \wedge b \times f \times e \wedge b \times h \times r)}{\mathbf{P}(a \times h \times r)} \\[2ex]
\dfrac{\text{R2} \wedge (b \times g \times w \wedge c \times g \times o \wedge c \times h \times r)}{\mathbf{P}(b \times h \times r)} \\[2ex]
\dfrac{\text{R2} \wedge (b \times g \times w \wedge c \times g \times o \wedge c \times i \times r)}{\mathbf{P}(b \times i \times r)} \\[2ex]
\dfrac{\text{R2} \wedge (b \times g \times w \wedge c \times g \times o \wedge c \times h \times w)}{\mathbf{P}(b \times h \times w)}
\end{array} \qquad (36)$$

De la même manière, on peut voir que:

$$\frac{R1 \wedge (b \times f \times e \wedge a \times f \times o)}{b \times i \times r \Rightarrow \mathbf{P}(a \times i \times r)} \quad (37)$$

On peut d'ailleurs noter que, de manière générale, l'utilisation des règles de résolution de la logique propositionnelle classique à l'intérieur d'un monde permet de diminuer notablement la taille des tableaux générés, ce qui est un point important dans le cas d'une automatisation de la méthode. Une variante de la méthode des tableaux incluant de telles améliorations est présentée dans [Catach 1991].

Le tableau obtenu en appliquant la méthode présentée précédemment au 3.1.2.3.1 à la spécification considérée est présenté dans la figure 22.

<1> **P**(a×h×r)
<1> **P**(b×h×r)
<1> **P**(b×i×r)
<1> **P**(b×h×w)
<1> b×i×r ⇒ **P**(a×i×r)
<1,1> a×h×r
<1,2> b×h×r
<1,3> b×i×r
<1,4> b×h×w

<1> (b×i×r)

<1,3> **P**(a×i×r)
<1,3,1> a×i×r

Figure 22 - Tableau obtenu

On peut identifier alors la structure d'un modèle permettant d'invalider la propriété P1 dans ce tableau. Ce modèle est présenté dans la figure 23.

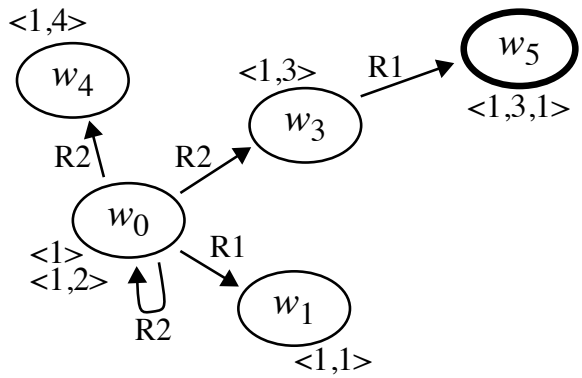


Figure 23 - Modèle invalidant la propriété P1

Ce contre-exemple permet de prouver que la propriété P1 n'est pas garantie. Une démarche analogue permettrait de constater que la propriété P2 est, elle, obtenue. En effet, une application exhaustive des règles permet de constater qu'il n'existe pas de contre-exemple permettant de la réfuter (le tableau complet n'est pas présenté ici). Ces résultats correspondent à ceux obtenus dans [Dacier 1994].

3.1.3 Évaluation quantitative

La méthode d'évaluation quantitative que nous pouvons utiliser à partir du graphe des privilèges (ou du processus d'intrusion) est celle définie dans [Dacier 1994; Dacier *et al*. 1996] qui a été présentée au 1.3.3.3 (page 50).

Une évaluation quantitative du niveau de sécurité courant du système peut permettre à un administrateur de sécurité d'identifier les failles du système qui peuvent être éliminées en apportant une amélioration significative de la sécurité sans avoir d'incidences notables sur les fonctions du système. Cette évaluation permet aussi de surveiller l'évolution de la sécurité opérationnelle du système, notamment en fonction des modifications survenant dans l'environnement, dans la configuration, dans les applications ou dans le comportement des utilisateurs.

Les mesures obtenues grâce à des outils d'évaluation doivent représenter aussi précisément que possible la sécurité du système en vie opérationnelle, c'est-à-dire sa capacité à résister à d'éventuelles attaques, ou encore de manière équivalente, la difficulté pour un attaquant d'exploiter les vulnérabilités présentes dans le système pour mettre en défaut les objectifs de sécurité. Ces définitions permettent d'identifier plusieurs caractéristiques attendues pour ces mesures:

- La mesure de la sécurité doit caractériser la sécurité du système lui-même indépendamment des menaces auxquelles il peut être confronté: le système est le même (et sa mesure de sécurité devrait être la même) qu'il soit confronté à un ou plusieurs attaquants possédant peu ou beaucoup d'expérience et de ténacité. Bien entendu, la sécurité d'un système donné a plus de chances d'être mise en défaut par plusieurs attaquants très compétents qui coopèrent que par un seul attaquant sans expérience.
- La mesure de la sécurité est directement reliée aux objectifs de sécurité: un système est sûr aussi longtemps que ses principaux objectifs de sécurité sont atteints, même s'il est relativement facile d'effectuer un certain nombre d'actions illégitimes qui ne mettent pas en défaut ces objectifs. Par exemple, un système est sûr même s'il est facile pour un intrus extérieur d'accéder à des informations publiques.
- La mesure de la sécurité doit évoluer en fonction des modifications effectuées dans le système qui influencent sa sécurité: toute modification peut engendrer de nouvelles vulnérabilités, ou faire disparaître des vulnérabilités préexistantes. La mesure de la sécurité doit donc être sensible à ces modifications. La principale vocation de cette mesure est de permettre de surveiller l'évolution de la sécurité d'un système donné, plutôt que de comparer le niveau absolu de sécurité de différents systèmes. Du point de vue opérationnel, on peut considérer

qu'il est plus important de savoir si la sécurité d'un système donné se dégrade ou s'améliore que de comparer des systèmes indépendants, possédant des objectifs, des utilisateurs, et un environnement différents. Par ailleurs, on peut considérer que le niveau de sécurité du système dépend également de la structure et de la nature du système ainsi que de son contenu. La mesure de sécurité dépend donc du système. Sa signification absolue, même si elle peut avoir un sens vis-à-vis d'un système considéré à un instant donné, ne permet donc pas de comparer deux systèmes différents.

Nous présentons dans les sections suivantes un certain nombre de mesures qui peuvent être définies à partir d'une représentation des vulnérabilités du système sous forme de graphe des privilèges, et dont les propriétés correspondent à ces définitions.

3.1.3.1 Hypothèses de comportement

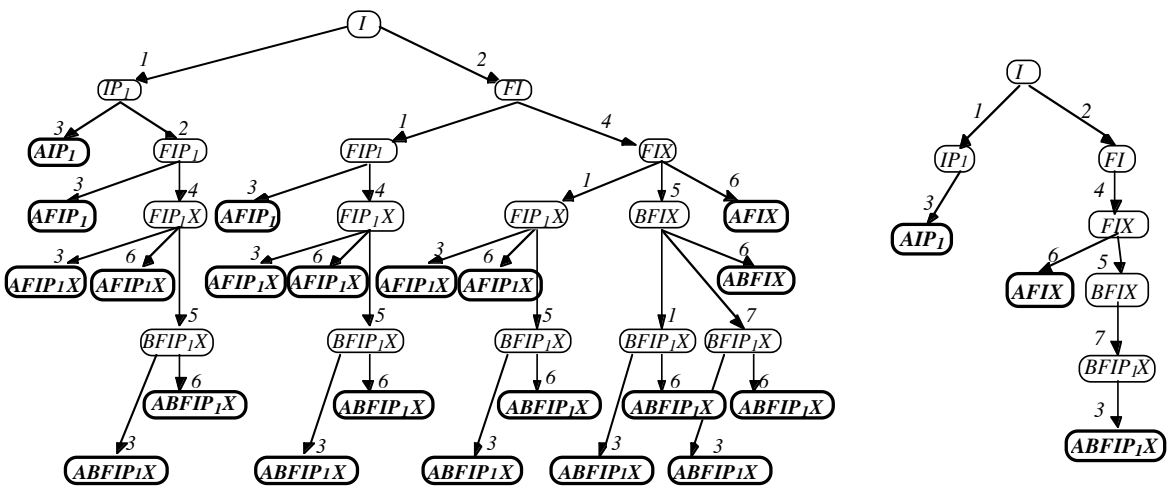
Afin d'évaluer les mesures quantitatives basées sur le graphe des privilèges qui caractérisent la sécurité opérationnelle, il est d'abord nécessaire d'identifier les scénarios d'attaque qui pourraient être suivis par un attaquant potentiel cherchant à atteindre sa cible. Certaines hypothèses fondamentales sur le comportement de l'attaquant doivent être définies afin d'obtenir ces différents scénarios. Tout d'abord, nous supposons que l'attaquant est sensé, et qu'il n'essaiera pas de mettre en œuvre une attaque qui lui permettrait d'obtenir des privilèges qu'il possède déjà. Des hypothèses supplémentaires sont nécessaires pour caractériser la progression de l'attaquant vers la cible. Différents modèles peuvent être définis en se basant sur ces hypothèses concernant le comportement de l'attaquant. Dans le premier modèle considéré, nous faisons l'hypothèse que l'attaquant choisit le plus court chemin qui le mène à la cible (noté SP pour "*Shortest Path*" par la suite), c'est-à-dire celui qui possède la valeur d'effort moyen cumulé la plus faible. Le plus court chemin peut être évalué directement à partir du graphe des privilèges et des valeurs associées à ses arcs. Cependant, cette hypothèse signifie implicitement que l'attaquant connaît à l'avance toute la topologie du graphe. Mais, pour construire le graphe des privilèges complet, l'attaquant doit posséder l'ensemble des privilèges mentionnés dans le graphe, et si l'attaquant possède déjà ces privilèges, il n'a aucun besoin de mettre en œuvre une attaque ! Clairement, l'hypothèse que l'attaquant emprunte le plus court chemin vers sa cible n'est pas satisfaisante. Dans la suite, nous introduisons deux nouvelles hypothèses et nous montrons que les mesures de sécurité correspondantes apportent plus d'information à un administrateur de sécurité que la mesure SP.

L'accroissement des privilèges de l'attaquant au fur et à mesure de sa progression vers la cible peut être caractérisé par un graphe d'état (appelé le **processus d'intrusion**) où chaque état identifie les privilèges qui ont été obtenus, et où les transitions entre états ont lieu quand l'attaquant réussit à exploiter une vulnérabilité qui lui permet d'acquérir de nouveaux privilèges. Afin de caractériser pleinement le processus d'intrusion obtenu à partir du graphe des privilèges, nous devons faire une hypo-

thèse supplémentaire qui définit quelles attaques vont être tentées par un attaquant à chaque étape du processus d'attaque. Deux hypothèses différentes sont présentées ci-après, chacune d'elles correspondant à un modèle spécifique du processus d'attaque (et donc à un comportement de l'attaquant):

- L'hypothèse de *mémoire totale* (MT): à chaque étape du processus d'attaque, toutes les possibilités d'attaques sont considérées, c'est-à-dire à la fois celles disponibles grâce aux privilèges nouvellement acquis et celles déjà utilisables à partir de privilèges précédemment obtenus mais que l'attaquant n'avait pas exploitées jusqu'alors. À chaque étape, l'attaquant peut donc choisir une attaque parmi l'ensemble de toutes les attaques qui lui sont possibles.
- L'hypothèse de *mémoire locale* (ML): à chaque fois que l'attaquant obtient un nouveau privilège, c'est-à-dire à chaque fois qu'il atteint un nouveau nœud du graphe des privilèges, il choisit de mettre en œuvre une attaque parmi celles qui peuvent être effectuées à partir de ce nœud (et du privilège associé) *uniquement*, sans considérer les autres attaques qui auraient pu être effectuées grâce à des privilèges précédemment acquis.

La figure 24 présente les processus d'intrusion associés au graphe des privilèges donné dans la figure 4 (page 51) quand les hypothèses MT et ML sont considérées. On peut voir que les scénarios d'attaques représentés dans la figure 24-b correspondent à un sous-ensemble de ceux identifiés dans la figure 24-a (identique à la figure 6, page 53).



(a) Hypothèse MT **(b)** Hypothèse ML

Figure 24 - Comparaison des processus d'intrusion associés au graphe des privilèges de la figure 4

3.1.3.2 Mesures

Afin de comparer l'évolution des mesures de sécurité correspondant aux hypothèses MT, ML et SP, on doit préciser le modèle mathématique utilisé pour évaluer l'effort moyen nécessaire à un attaquant pour atteindre la cible. À l'instar de [Dacier 1994;

Dacier *et al*. 1996], nous utilisons un modèle Markovien qui satisfait un certain nombre de propriétés intuitives concernant l'évolution de la sécurité. D'autres approches peuvent être envisagées pour obtenir une évaluation quantitative de cet effort, par exemple en assimilant les nœuds du graphe des privilèges à un domaine de protection à l'instar de [Moskowitz & Kang 1997]. Le modèle Markovien est basé sur l'hypothèse que la probabilité de succès d'une attaque élémentaire avant qu'une quantité d'effort e soit dépensée est décrite par une distribution exponentielle donnée par $P(e) = 1 - \exp(-\lambda e)$, où λ désigne le taux de succès associé à l'attaque. L'utilisation de cette distribution implique que:

- un attaquant potentiel réussira à atteindre la cible, si un chemin menant à cette cible existe et s'il dépense un effort suffisant;
- l'effort moyen nécessaire au succès d'une attaque donnée est égal à $1/\lambda$.

Le dernier point est particulièrement remarquable car il signifie que la connaissance des taux moyens de succès des attaques est suffisant pour caractériser toute la distribution. Le premier point demande quelques éclaircissements. En fait, notre objectif étant d'évaluer la résistance du système à des attaques visant une cible particulière, nous considérons seulement les scénarios d'attaques qui mènent à cette cible et non les scénarios qui pourraient échouer.

Chaque transition dans le processus d'intrusion est associée aux taux de succès de la vulnérabilité correspondante. Plusieurs mesures probabilistes peuvent être définies sur ce modèle, parmi elles, l'effort moyen nécessaire à un attaquant potentiel pour atteindre la cible, noté METF (*Mean Effort To security Failure*, par analogie avec le *Mean Time To Failure*). Cette mesure permet une interprétation simple des résultats: plus le METF est élevé, meilleure est la sécurité. De plus, des expressions analytiques simples peuvent être obtenues et analysées afin de vérifier la plausibilité des résultats fournis par le modèle.

Le METF est donné par l'expression (38) déjà mentionnée au 1.3.3.3 (page 52).

$$MTTF_k = T_k + \sum_{l \in out(k)} P_{kl} \times MTTF_l \quad \text{où} \quad T_k = \frac{1}{\sum_{l \in out(k)} \lambda_{kl}} \text{ et } P_{kl} = \lambda_{kl} \times T_k \tag{38}$$

Il apparaît clairement que la complexité de l'algorithme de calcul du METF est directement liée à la taille du processus d'intrusion, or pour l'hypothèse MT, le nombre de chemins à considérer est beaucoup plus important que pour l'hypothèse ML.

3.1.3.3 Comportements attendus

Suivant que les hypothèses MT, ML ou SP sont considérées, on peut espérer un comportement différent du METF.

Considérons l'exemple du graphe des privilèges ne contenant qu'un seul chemin entre le nœud de l'attaquant et celui de la cible (figure 25). Dans ce cas, le METF est donné par $\sum_{j=1\ldots k} \frac{1}{\lambda_j}$ où k est le nombre d'arcs figurant dans le chemin, et λ_j est

le taux de succès associé à la vulnérabilité élémentaire *j*. La même valeur est obtenue quelle que soit l'hypothèse MT, ML ou SP considérée. Clairement, quand le nombre d'arcs augmente, le METF augmente et la sécurité s'améliore. On observe aussi que, quand la valeur de λ_j augmente, le METF diminue et la sécurité se dégrade.

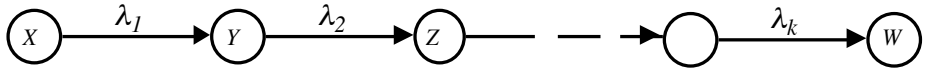
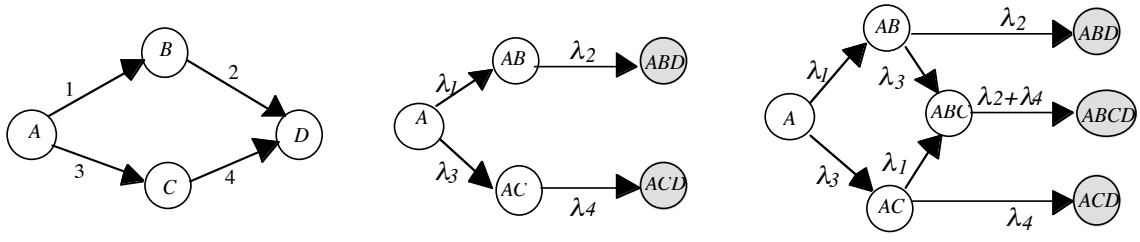


Figure 25 - Modèle de Markov correspondant à un seul chemin

Dans le cas où le graphe des privilèges révèle plusieurs chemins entre le nœud identifiant l'attaquant et celui identifiant la cible, le plus court chemin est obtenu en identifiant dans le graphe des privilèges tous les chemins directs qui vont de l'attaquant à la cible, puis en recherchant la valeur minimale du METF parmi toutes celles calculées pour ces chemins directs. Un chemin direct entre l'attaquant et la cible est tel que chaque nœud appartenant à ce chemin est visité au plus une fois. L'expression de METF_{SP} est donc : $\text{METF}_{\text{SP}} = \min\{u_1, ..., u_n\}$ où $u_k = \sum_{i = 1 \ldots l(k)} 1/\lambda_i$, λ_i est le taux correspondant à l'arc *i* appartenant au chemin direct *k* qui en compte $l(k)$, et *n* est le nombre de chemins directs.

Les valeurs du METF correspondant aux hypothèses MT ou ML peuvent être obtenues à partir du processus d'intrusion. Considérons l'exemple de la figure 26 où *A* est l'attaquant et *D* la cible. Le graphe des privilèges (figure 26-a) indique la présence de deux chemins menant à la cible. Le modèle de Markov correspondant aux hypothèses ML et MT est donné dans les figure 26-b et 26-c respectivement.



(a) Graphe des privilèges **(b)** Hypothèse ML **(c)** Hypothèse MT

Figure 26 - Chemins multiples – exemple

L'application de (38) conduit alors aux expressions suivantes:

$$\begin{aligned} \mathrm{METF}_{\mathrm{ML}} &= \frac{1}{\lambda_1+\lambda_3} + \frac{\lambda_1}{\lambda_1+\lambda_3} \times \frac{1}{\lambda_2} + \frac{\lambda_3}{\lambda_1+\lambda_3} \times \frac{1}{\lambda_4} \\ \mathrm{METF}_{\mathrm{MT}} &= \frac{1}{\lambda_1+\lambda_3} + \frac{\lambda_1}{\lambda_1+\lambda_3} \times \left(\frac{1}{\lambda_2+\lambda_3} + \frac{\lambda_3}{\lambda_2+\lambda_3} \times \frac{1}{\lambda_2+\lambda_4}\right) + \\ & \quad \frac{\lambda_3}{\lambda_1+\lambda_3} \times \left(\frac{1}{\lambda_1+\lambda_4} + \frac{\lambda_1}{\lambda_1+\lambda_4} \times \frac{1}{\lambda_2+\lambda_4}\right) \end{aligned} \tag{39}$$

On peut voir que, pour chaque valeur de λ_1, λ_2, λ_3 et λ_4, l'expression de $\mathrm{METF}_{\mathrm{MT}}$ est toujours inférieure à $1/\lambda_1 + 1/\lambda_2$ (qui correspond au cas où seul le premier chemin existe), et à $1/\lambda_3 + 1/\lambda_4$ (qui correspond au cas où seul le second chemin existe). Ce résultat illustre le fait que l'apparition de nouveaux chemins permettant d'atteindre la cible dans le graphe des privilèges conduit à une diminution de $\mathrm{METF}_{\mathrm{MT}}$ qui indique une dégradation de la sécurité. Ce résultat est généralisable [Dacier 1994].

L'hypothèse ML conduit à un comportement différent car $\mathrm{METF}_{\mathrm{ML}}$ peut augmenter ou diminuer suivant la valeur des paramètres. Par exemple, $\mathrm{METF}_{\mathrm{ML}}$ n'est inférieur à $1/\lambda_1 + 1/\lambda_2$ que si $1/\lambda_4 < 1/\lambda_1 + 1/\lambda_2$, c'est-à-dire quand l'effort moyen nécessaire pour obtenir les privilèges du nœud *D* à partir du nœud *C* est inférieur à l'effort moyen correspondant au chemin complet passant par *B*. Ceci est dû au fait que, en faisant l'hypothèse ML et contrairement à l'hypothèse MT, quand l'attaquant choisit un chemin donné, il ne revient jamais en arrière jusqu'à ce qu'il ait atteint la cible. Si des modifications introduites dans le graphe des privilèges conduisent à des chemins additionnels qui sont plus courts que ceux du graphe des privilèges initial, alors la valeur de $\mathrm{METF}_{\mathrm{ML}}$ diminue, sinon la valeur de $\mathrm{METF}_{\mathrm{ML}}$ augmente. (Tandis que dans tous les cas, l'apparition de chemins additionnels entraîne une diminution de $\mathrm{METF}_{\mathrm{MT}}$.)

3.1.3.4 Discussion

Le tableau 8 résume les comportements attendus des mesures $\mathrm{METF}_{\mathrm{ML}}$ et $\mathrm{METF}_{\mathrm{MT}}$ quand le nombre de chemins entre l'attaquant et la cible augmente.

Notons que nous ne considérons pas l'avènement de plusieurs modifications *simultanées* du graphe des privilèges (c'est-à-dire l'ajout ou la disparition de plusieurs vulnérabilités, ou la modification des taux associés à plusieurs vulnérabilités). Des modification simultanées peuvent influencer différemment la sécurité du système et le comportement des mesures observées dans ce cas peut être quelconque.

Si une seule modification est apportée au graphe des privilèges, nous devrions observer que:

- si le nombre de chemins augmente (en raison de l'apparition d'une nouvelle vulnérabilité), $METF_{MT}$ diminue puisque ce nouveau chemin affaiblit la sécurité de la cible;
- quand la longueur du plus court chemin entre l'attaquant et la cible diminue, $METF_{MT}$ diminue et montre une dégradation de la sécurité;
- ainsi qu'il a été précisé dans la section précédente, deux types de comportements peuvent être observés pour $METF_{ML}$:
  - si le nouveau chemin diminue la probabilité d'emprunter un chemin existant relativement court au profit d'un chemin plus long, $METF_{ML}$ peut augmenter (comportement 2 dans le tableau 8);
  - sinon, $METF_{ML}$ doit avoir le même comportement que $METF_{MT}$: la mesure doit diminuer et indiquer une dégradation de la sécurité quand un nouveau chemin apparaît (comportement 1).

Nombre de chemins	$METF_{MT}$	$METF_{ML}$	Comportement
↑	↓	↓	1
↑	↓	↑	2

Tableau 8 - Types de comportement attendus

Il apparaît donc que l'hypothèse MT est celle qui permet l'interprétation la plus directe de l'évolution de la sécurité. En analysant les variations de $METF_{MT}$ en même temps que les modifications apparues dans le graphe des privilèges entre deux observations successives, l'administrateur de sécurité peut déterminer si ces modifications ont eu un impact significatif sur la sécurité. En se basant sur ces résultats, il peut identifier les chemins les plus critiques du graphe des privilèges et prendre les mesures appropriées: soit corriger certaines des vulnérabilités du système (quand la sécurité diminue notablement) soit maintenir la configuration courante du système dans le cas où les risques entraînés par les modifications observées ne sont pas significatifs (c'est-à-dire si la sécurité augmente, ou si seule une faible diminution du METF a été observée).

En ce qui concerne l'hypothèse ML, une augmentation de $METF_{ML}$ quand le nombre de chemins augmente peut être considérée comme un comportement discutable. En effet, ceci semble signifier que la sécurité peut augmenter quand de nouvelles vulnérabilités sont introduites dans le système. Ces évolutions sont liées au comportement de l'attaquant dans l'hypothèse ML qui suppose que, quand l'attaquant choisit un chemin donné, il ne le quitte pas. Si les premières vulnérabilités apparaissant dans le chemin sélectionné correspondent à des attaques faciles, l'attaquant est plus enclin à les emprunter. Pourtant, si la mise en œuvre des attaques suivantes demande beaucoup plus d'effort, l'effort moyen nécessaire pour atteindre la cible va augmenter. Un problème délicat est donc de savoir si cette hypothèse correspond

à un comportement réaliste de l'attaquant. Il est très difficile de répondre directement à cette question étant donné l'absence de données concrètes sur le comportement des attaquants. Dans la mise en œuvre expérimentale de la méthode présentée dans la section suivante (cf 3.2.1) nous montrons que des informations pertinentes sur l'évolution de la sécurité peuvent être fournies à un administrateur de sécurité même quand le modèle associé à ML est utilisé. En fait, étant donné que ce sont essentiellement les variations du METF plutôt que la valeur absolue de la mesure qui sont étudiées, nous verrons que toute variation du METF doit être examinée (ne serait-ce que parce que dans le cadre d'une observation réelle il n'est pas possible de savoir a priori en observant l'évolution de la mesure si une ou plusieurs modifications sont survenues dans le graphe des privilèges).

En ce qui concerne la longueur du plus court chemin, il est évident que l'information fournie par cette mesure est incomplète étant donné qu'elle ne prend en compte qu'un seul des chemins existants dans le graphe des privilèges. Une variation de la sécurité due à la présence d'autres chemins dans le graphe ne pourra donc pas être identifiée si cette mesure est utilisée seule pour surveiller la sécurité opérationnelle du système.

3.1.3.5 Validité des mesures

La méthode d'évaluation quantitative présentée dans ce mémoire permet de définir un certain nombre de mesures. Toutefois, une question majeure est soulevée par la présentation théorique de cette approche : ces mesures sont-elles valides et permettent-elles de représenter fidèlement la sécurité du système ? Dans ce domaine, une validation directe est difficile : les attaques réelles sur des systèmes réels sont trop rares pour que l'on puisse chercher une corrélation précise entre les mesures calculées et le taux de succès de ces attaques. Même une approche comme celle des "*Tiger Teams*" resterait probablement inefficace étant donné que les attaques mises en œuvre dans ce cadre ne sont pas forcément représentatives d'attaques réelles. De plus, les attaques observées doivent être effectuées sur un système stable, et en très grand nombre, pour permettre d'atteindre une bonne précision [Olovsson *et al*. 1995], tandis que les mesures présentées précédemment sont destinées à surveiller l'évolution dynamique de la sécurité du système. Dans ce cas, la seule validation possible reste une validation expérimentale. Il est donc nécessaire d'observer l'évolution des mesures calculées à partir de l'observation d'un système réel de grande taille pendant une longue période, et de déterminer les événements qui sont à l'origine de chaque variation significative de leurs valeurs.

Dans un système réel, nous verrons qu'il est fréquent d'observer des modifications élémentaires du graphe des privilèges entre deux observations successives. Ceci permet tout d'abord de vérifier que les variations observées correspondent bien, pour chacune des mesures, aux comportements attendus tels qu'ils ont été décrits précédemment (cf 3.1.3.3). Ensuite, l'étude de la pertinence vis-à-vis de la sécurité

des événements (éventuellement multiples) survenus à chaque variation significative des valeurs obtenues nous permettra d'apprécier le niveau de confiance que l'on peut accorder aux mesures quantitatives de la sécurité que nous avons définies.

3.2 Mise en œuvre

Afin d'illustrer la méthode d'évaluation présentée précédemment, nous envisageons à présent son application dans deux cadres distincts. Tout d'abord, nous présentons l'utilisation de cette méthode dans le cadre de la surveillance de la sécurité d'un système informatique de grande taille sur une longue durée. Cette expérience nous permet de vérifier la pertinence des mesures quantitatives de la sécurité définies précédemment, puis de les comparer. Ensuite, nous présentons les résultats de l'évaluation quantitative appliquée à l'étude d'une organisation réelle.

3.2.1 Application à un système informatique : UNIX

Dans le cadre d'une application expérimentale dans une organisation, l'observation du système permettant d'obtenir une évaluation quantitative de sa sécurité peut difficilement être menée en continu. Ceci est principalement dû au temps et aux efforts nécessaires pour étudier complètement le fonctionnement sur le terrain, cette étude devant prendre en compte toutes les opérations effectuées et l'ensemble des vulnérabilités considérées. Dans le cas d'un système informatique, l'observation d'un nombre important de vulnérabilités est rendu plus aisée par la possibilité d'analyser le système grâce à des outils automatiques. De tels outils permettent notamment d'effectuer une observation régulière du système pendant sa vie opérationnelle afin d'étudier l'évolution des mesures quantitatives de sécurité. Nous présentons dans cette section les résultats de la mise en œuvre de la méthode d'évaluation dans un système UNIX de grande taille pendant une longue période, et les conclusions qui en découlent concernant la pertinence des mesures quantitatives de la sécurité que nous avons définies.

3.2.1.1 Vulnérabilités étudiées

L'évaluation des mesures de sécurité nécessite la définition d'un ensemble d'attaquants et de cibles. Ces paires sont reliées aux objectifs de sécurité du système. Pour un système UNIX, une des cibles principales à protéger est le compte du super-utilisateur `root`. Une autre cible intéressante à étudier est le groupe constitué par l'ensemble des administrateurs du système, qui donne accès aux différentes données qu'ils se partagent. Pour sélectionner un attaquant précis, nous avons

choisi l'utilisateur `insider` qui représente les privilèges minimaux possédés par tout utilisateur du système (par exemple, le privilège de se connecter, de changer son mot de passe, etc.). Le tableau 9 résume ces différents cas.

	Attaquant	Cible
Objectif 1	`insider`	`root`
Objectif 2	`insider`	`admin_group`

Tableau 9 - Objectifs de sécurité

Pour l'analyse du second objectif de sécurité, un problème spécifique à UNIX doit être mentionné. Ce problème est dû à l'existence de super-utilisateurs sous UNIX. Un super-utilisateur est en mesure d'obtenir tous les privilèges des autres utilisateurs dans le système. Donc, si on considère une cible constituée par un ensemble d'utilisateurs normaux, ce mécanisme nous conduit implicitement à inclure le super-utilisateur dans cet ensemble (puisque cet ensemble de privilèges inclut tous les autres). Dans ce cas, l'objectif 2 implique l'objectif 1, au sens où si une séquence de méthodes de transfert de privilèges permet de mettre en défaut l'objectif 1, elle permet aussi de mettre en défaut l'objectif 2. Pour avoir des cas d'étude totalement distincts, nous n'avons pas considéré les vulnérabilités liées aux propriétés du super-utilisateur pour l'étude de l'objectif 2. Nous avons donc retiré du graphe des privilèges tous les arcs allant immédiatement du super-utilisateur vers les autres ensembles de privilèges.

Parmi toutes les catégories de vulnérabilités connues sous UNIX, 13 parmi les plus communes ont été observées. Ceci inclut: la vérification des mots de passe; les méthodes de transfert de privilèges configurables par les utilisateurs (`.rhosts`); les permissions incorrectes sur les fichiers `setuid`, `.rhosts` et les fichiers d'initialisation; la configuration incorrecte de la recherche automatique de commandes qui permet les attaques par chevaux de Troie; etc. Une présentation plus détaillée des différentes vulnérabilités présentes dans UNIX pourra être trouvée dans [Garfinkel & Spafford 1996].

Des modifications de l'état de sécurité, ou *événements* de sécurité, ont lieu quand des vulnérabilités sont soit créées soit éliminées (des arcs sont ajoutés ou retirés du graphe des privilèges) ou bien quand la valeur associée à une vulnérabilité (le taux associé à un arc) change. De tels événements ont pu être observés fréquemment pendant l'expérience.

Dans le cadre de l'expérience, nous avons défini une échelle de classification des vulnérabilités comptant quatre niveaux présentée dans le tableau 10. Les valeurs associées à chacun des niveaux diffèrent d'un ordre de grandeur et correspondent aux taux de succès associés aux différentes vulnérabilités élémentaires : `niveau1` correspond aux attaques élémentaires les plus faciles, et `niveau4` aux plus difficiles.

Nom	Valeur
`niveau1`	10^{-1}
`niveau2`	10^{-2}
`niveau3`	10^{-3}
`niveau4`	10^{-4}

Tableau 10 - Taux de succès

Les différents niveaux associés à chacune des méthodes d'attaque étudiées sont relativement arbitraires. L'évaluation précise du taux de succès des différentes vulnérabilités présentes dans le système nécessiterait l'utilisation d'outils additionnels (par exemple pour construire un profil des utilisateurs) qui ne sont pas disponibles dans le prototype utilisé. Toutefois, ceci ne constitue pas un inconvénient majeur dans notre expérience, puisque l'objectif est avant tout de valider le comportement et la pertinence des mesures de sécurité étudiées.

3.2.1.2 Présentation du système cible

Le système observé dans cette expérience est un système informatique distribué de grande taille comptant plusieurs centaines de stations de travail connectées à un réseau local. Il compte environ 700 utilisateurs qui partagent un même système de fichiers global (NFS). Pendant l'expérience, le nombre total d'utilisateurs a fréquemment varié, notamment en raison de l'arrivée ou du départ d'utilisateurs temporaires (un événement courant dans le système cible). La recherche des vulnérabilités du système est effectuée dans le système de fichiers partagé. Dans cette expérience, le système a été observé pendant 13 mois sur une base journalière, à partir de juin 1995 et jusqu'à la fin de juillet 1996. L'archive des graphes de privilèges contenait 385 éléments à cette date (un pour chaque jour en général).

Dans ce système, la sécurité n'est pas la principale préoccupation des utilisateurs. Comme aucune information sensible n'est stockée dans le système, aucune politique de sécurité exigeante n'est imposée, même si les administrateurs du système ou certains de ses utilisateurs se préoccupent parfois de la sécurité pour des raisons personnelles, ou pour des raisons de sécurité-innocuité. Ceci explique le nombre important de vulnérabilités qui ont été observées dans le système. Ces vulnérabilités sont connues, persistent longtemps dans le système, et sont tolérées parce qu'elles correspondent généralement à des fonctionnalités commodes ou utiles.

Enfin, notre objectif principal étant de valider le comportement des mesures de sécurité utilisées, nous nous sommes contentés d'observer l'évolution naturelle du système. Aucune tentative n'a été faite pour convaincre les utilisateurs de retirer certaines des vulnérabilités identifiées ce qui aurait pu améliorer la sécurité du système.

3.2.1.3 Description du prototype

L'expérience présentée dans cette section a été mise en œuvre à l'aide d'un certain nombre d'outils. L'objectif principal de ces outils prototypes est de fournir des éléments expérimentaux permettant de valider la pertinence des mesures quantitatives de la sécurité proposées. Les principales étapes de l'implémentation du processus d'évaluation sont:

1) *Définition de la politique de sécurité*: Pour chacun des objectifs de sécurité choisis pour le système, les cibles pertinentes (c'est-à-dire les ensembles de privilèges qui doivent être protégés) et les attaquants potentiels (c'est-à-dire les ensembles de privilèges vis-à-vis desquels les cibles doivent être protégées) sont identifiés. Chaque paire cible-attaquant correspond à deux ensembles de nœuds dans le graphe des privilèges, pour lesquels une évaluation quantitative de la sécurité doit être effectuée. Un outil disposant d'une interface graphique a été développé, il permet de décrire formellement les objectifs de sécurité à partir desquels les différents nœuds du graphe sont identifiés [Laffont & Ortalo 1997]. Dans le cadre de l'étude des différents objectifs de sécurité présentés dans le tableau 9, l'utilisation d'un tel outil n'est pas indispensable. Mais l'objectif de l'implémentation d'un éditeur graphique de politiques de sécurité dans le contexte du prototype était de montrer la faisabilité et les apports de ce type d'outil. Dans le contexte d'une organisation, il semble en effet désirable de fournir un tel outil pour assister l'administrateur de la sécurité dans la définition de la politique de sécurité à l'aide d'un langage logique comme celui présenté au 2.2.3 (page 66).
2) *Analyse du système et construction du graphe des privilèges*: Nous avons développé un outil, nommé ASA pour Automatic Security Advisor, qui recherche dans le système UNIX analysé la présence d'un ensemble de vulnérabilités connues. Cet outil construit alors le graphe des privilèges correspondant en s'appuyant sur un moteur d'inférence Prolog. L'outil d'analyse fonctionne avec des privilèges étendus, de manière à pouvoir analyser tous les éléments du système. Dans la version utilisée, ASA exploite certaines des procédures déjà incluses dans l'outil COPS [Farmer & Spafford 1994]. Plus précisément, comme dans COPS, un certain nombre de scripts UNIX analysent le système de fichiers, rassemblant des informations concernant les droits d'accès de certains fichiers particuliers appartenant à chaque utilisateur, ou figurant dans des répertoires spécifiques. Le programme `crack` (v4.1) est exécuté en parallèle afin de tenter de deviner les mots de passe des utilisateurs à partir d'un dictionnaire standard. Chaque fois qu'une vulnérabilité est détectée dans le système, un arc est ajouté au graphe des privilèges en construction. Comme il est impossible de

savoir, à ce point de l'analyse, si la vulnérabilité identifiée est réellement une faille du point de vue de la sécurité, aucune action correctrice n'est effectuée. La sortie de l'outil ASA est donc un graphe des privilèges décrivant toutes les vulnérabilités du système cible au moment de l'analyse. Après cette recherche, le graphe des privilèges est enregistré dans une archive. Cette archive est régulièrement mise à jour grâce aux outils classiques d'exécution automatique fournis par UNIX, comme `cron`, qui permettent de lancer périodiquement une analyse automatique du système.

3) *Evaluation quantitative*: Par la suite, un autre outil calcule les mesures de sécurité présentées au 3.1.3 pour chacun des objectifs de sécurité. Ces calculs peuvent être effectués sur un seul graphe, ou sur toute une archive.

4) *Identification des événements pertinents*: Enfin, pour faciliter l'analyse des mesures de la sécurité qui ont été calculées, un outil identifie pour chaque variation significative de la mesure l'événement de sécurité qui en est la cause. Plus précisément, l'outil recherche les arcs, mentionnés dans les chemins entre l'attaquant et la cible, qui ont variés entre deux graphes des privilèges consécutifs. Ceci permet d'identifier les événements qui sont à l'origine de la variation de la mesure. Un exemple de la sortie de cet outil est présenté dans l'annexe B.

L'ensemble de ces différents outils constitue le prototype d'un logiciel d'évaluation quantitative de la sécurité, dénommé ÉSOPE, actuellement en cours de réalisation. Ce logiciel inclut les phases de spécification de la politique de sécurité (axée sur le contexte d'une organisation), d'analyse du système (axée sur l'observation automatique d'un système informatique), de calcul des mesures, et d'analyse des résultats obtenus. Tout comme la majeure partie des différents outils présentés précédemment, ÉSOPE a été réalisé dans le langage Perl [Wall *et al*. 1996; Dominus 1998].

3.2.1.4 Résultats

Les résultats de l'expérience menée à l'aide des outils présentés précédemment correspondant aux objectifs 1 et 2 sont présentés dans la figure 27 et la figure 28 respectivement. Les mesures présentées sont: le nombre de chemins identifiés entre l'attaquant et la cible, $METF_{SP}$, $METF_{ML}$ et $METF_{MT}$. La liste des événements de sécurité correspondants est donnée dans l'annexe B (page 163). Certains de ces événements sont également repérés dans les figures (et identifient des exemples de comportements particuliers, voir [Ortalo *et al*. 1997] pour plus de précisions).

La mesure $METF_{MT}$ ne peut être calculée que quand le nombre de chemins entre l'attaquant et la cible est relativement faible. Les lignes épaisses dans les deux graphiques présentent donc des vides correspondant à l'impossibilité de calculer cette mesure (malheureusement, ceci représente une longue période dans la figure 28). Chaque variation significative de la mesure indiquée sur ces courbes a été analysée, et une description détaillée de la cause de cette variation est fournie dans l'annexe B.

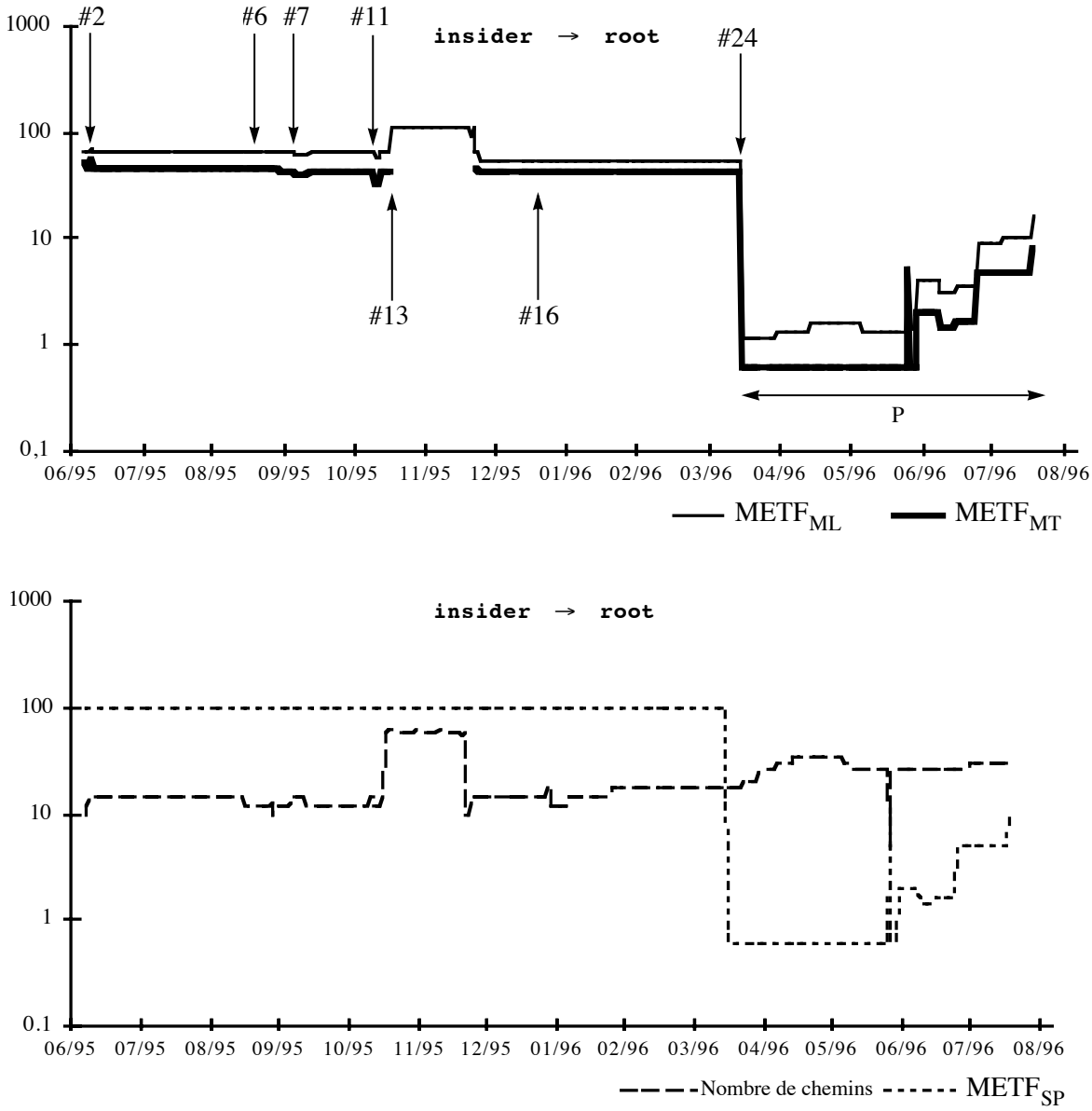


Figure 27 - Evolution des mesures (objectif 1)

3.2.1.5 Comparaison des différentes mesures

Durant toute la durée de l'expérience, la longueur du plus court chemin a évolué seulement un petit nombre de fois. Cette mesure fournit une information intéressante concernant le plus court chemin présent dans le système, toutefois elle ne présente guère d'intérêt du point de vue dynamique et n'est pas suffisamment sensible pour permettre de surveiller l'évolution de la sécurité. Comme nous l'avons indiqué précédemment, par rapport à $METF_{MT}$, la valeur du plus court chemin ne prend pas en compte le fait que plusieurs chemins équivalents puissent être disponibles.

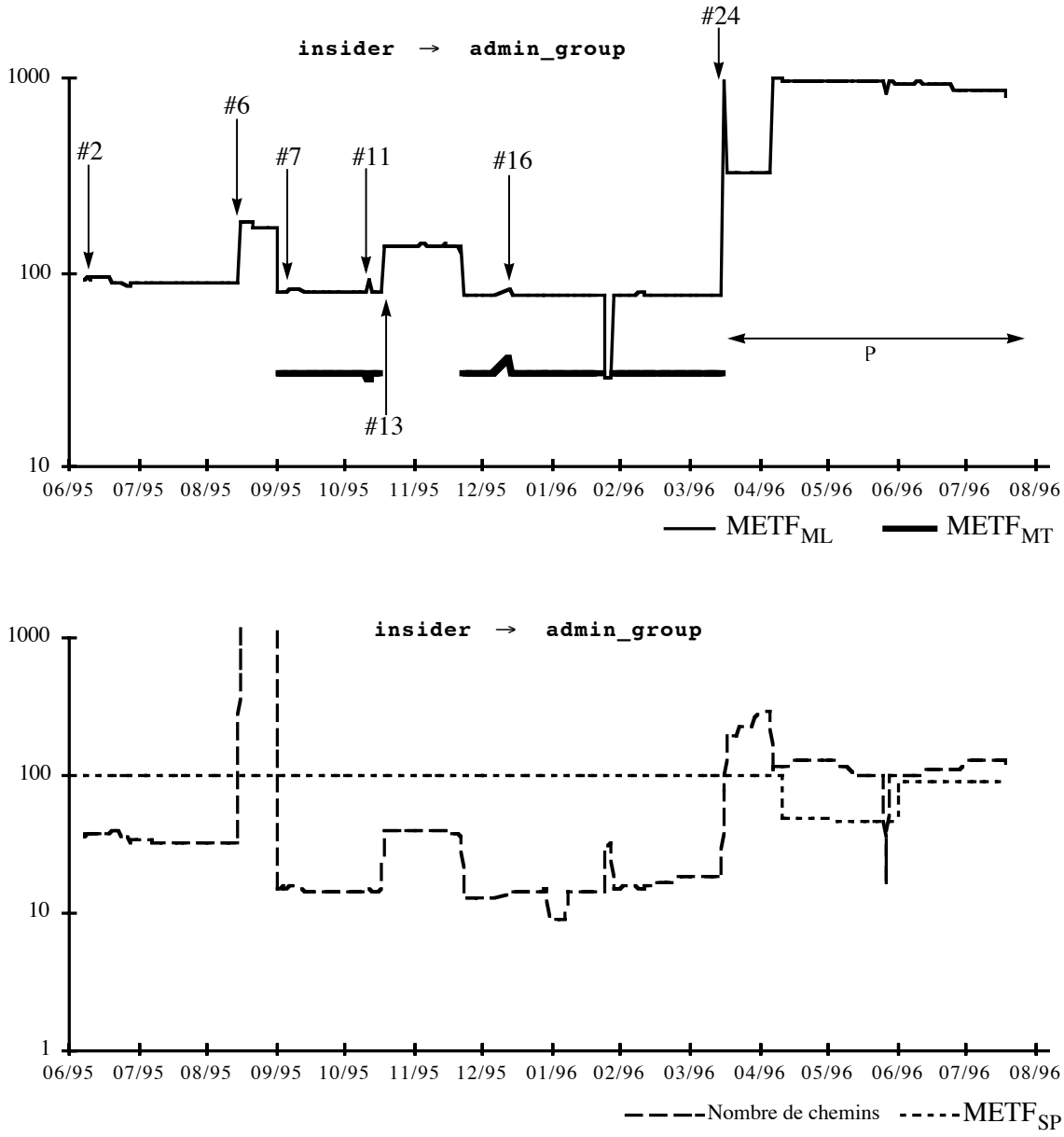


Figure 28 - Evolution des mesures (objectif 2)

En fait, plus que sa longueur, c'est la nature de ce chemin et les vulnérabilités correspondantes qui sont intéressantes pour améliorer la sécurité. En effet, ce chemin fait partie de ceux qui ont un impact majeur sur $METF_{MT}$ et $METF_{ML}$.

Le nombre de chemins existant entre l'attaquant et la cible est une information beaucoup plus sensible, mais qui semble difficile à utiliser directement pour surveiller les évolutions de la sécurité opérationnelle. En effet, on peut d'abord remarquer qu'un événement de sécurité conduisant à une augmentation ou une diminution du nombre de chemins ne conduit pas nécessairement à une variation significative des autres mesures de sécurité (voir par exemple les événements #1,18,19,20,22 du tableau 16, page 163). Au contraire, on peut identifier des évé-

nements qui conduisent à une évolution significative de $METF_{MT}$ ou $METF_{ML}$ tandis que le nombre de chemins varie peu (voir par exemple les événements #2,3,4,12,16,17,21,23 du tableau 16, page 163). Globalement, on peut donc voir que le nombre de chemins entre l'attaquant et la cible est une mesure qui déclencherait un nombre important d'alarmes parmi lesquelles certaines pourraient s'avérer relativement infondées. De plus, tous les événements de sécurité significatifs ne conduiraient pas forcément à une alarme. En conséquence, on peut considérer que cette mesure est moins fiable, et beaucoup plus difficile à utiliser que $METF_{MT}$ ou $METF_{ML}$.

Ces deux dernières mesures présentent un comportement intéressant, comportant des périodes de stabilité séparées par des variations significatives. Ainsi que l'on peut le voir dans l'analyse détaillée des événements, chaque variation de ces mesures peut être reliée à un événement pertinent du point de vue de la sécurité. Cependant, $METF_{MT}$ n'a pas pu être calculée dans tous les cas, ce qui est un inconvénient majeur. $METF_{ML}$ présente parfois un comportement délicat, montrant une augmentation de l'effort moyen nécessaire à l'attaquant pour atteindre la cible tandis que le nombre de chemins entre eux augmente (comportement 2 dans le tableau 8). Ceci diminue la confiance que l'on peut avoir dans $METF_{ML}$, d'autant plus qu'un seul événement de sécurité comme #6 ou #13 peut conduire à une augmentation importante de cette mesure. Toutefois, il semble possible d'avoir confiance dans $METF_{ML}$ pour révéler une dégradation de la sécurité de la cible, et pour réagir correctement aux événements de sécurité significatifs (contrairement au nombre de chemins).

De toutes les mesures étudiées, $METF_{MT}$ est celle qui présente le comportement le plus plausible. Des travaux complémentaires seraient nécessaires afin de réduire la complexité de l'algorithme utilisé pour la calculer, par exemple en se basant sur une technique de calcul approché, afin d'obtenir l'ensemble des valeurs manquantes.

3.2.1.6 Comparaison avec d'autres outils

Habituellement, les outils utilisés pour surveiller la sécurité opérationnelle d'un système, comme COPS [Farmer & Spafford 1990 ; Farmer & Spafford 1994] ou SATAN, se limitent à une analyse directe du système. Ils produisent donc une liste mentionnant différentes vulnérabilités identifiées dans le système, éventuellement regroupées dans différentes classes suivant leur sévérité. Le prototype utilisé pour obtenir les résultats présentés précédemment est largement appuyé sur ces outils. Il est donc possible de réunir les données qu'ils auraient produites s'ils avaient été utilisés seuls pendant l'expérience. Ceci nous permet d'effectuer une analyse comparative des informations qu'ils peuvent fournir avec celles obtenues après que l'évaluation quantitative ait été effectuée.

Ainsi, la figure 29 présente l'évolution du nombre total de vulnérabilités identifiées dans le système tout au long de l'expérience. La figure 30 présente les mêmes résultats, en détaillant la distribution des vulnérabilités recensées parmi les différents niveaux de sévérité utilisés (tableau 10).

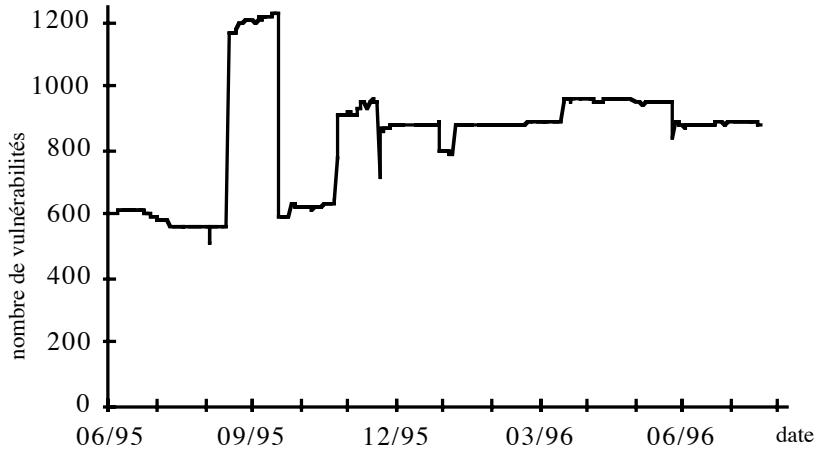


Figure 29 - Évolution du nombre total de vulnérabilités

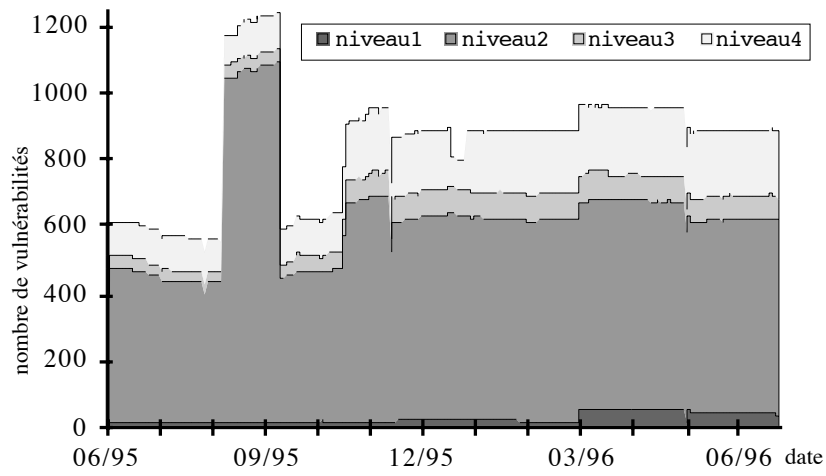


Figure 30 - Évolution de la distribution des vulnérabilités

On peut voir que l'utilisation directe des informations fournies par les figures 29 et 30 pour la surveillance de la sécurité du système conduit à un grand nombre d'alarmes. En fait, chaque fois qu'un événement de sécurité survient dans le système, l'administrateur de sécurité est obligé de l'analyser précisément, même s'il s'agit d'un événement mineur qui n'a pas d'influence sur les objectifs de sécurité. Pour diminuer le nombre d'alarmes, une solution simple consiste à prendre en compte le niveau de sévérité des nouvelles vulnérabilités identifiées. Néanmoins, comme on peut le voir en comparant la figure 30 aux figures 27 et 28, une évolution du nombre de vulnérabilités dangereuses (`niveau1` ou `niveau2`) et une diminution de la sécurité globale ne sont pas toujours corrélées. Ceci provient du fait que l'examen du seul nombre de vulnérabilités présentes dans le système ne tient pas compte de la manière dont l'exploitation successive de plusieurs vulnérabilités (de niveaux de sévérité différents) peut permettre de mettre en défaut un objectif de sécurité du système.

Bien évidemment, l'objectif de cette comparaison n'est pas de déprécier la valeur des résultats fournis par les outils classiques d'analyse de la sécurité d'un système UNIX comme COPS ou SATAN. La mise en œuvre d'une recherche automatique des faiblesses existant dans le système est une première étape essentielle pour la surveillance de la sécurité opérationnelle, et la méthode d'évaluation quantitative que nous présentons est largement appuyée sur les données fournies par de tels outils. Pourtant, et c'est un problème reconnu, les alarmes déclenchées par ce type d'outil sont très nombreuses, et il n'est pas facile pour l'administrateur de sécurité de les prendre en charge toutes systématiquement. Les mesures quantitatives permettent à l'administrateur d'extraire les événements qui demandent une intervention parmi tous ceux qui peuvent avoir été observés. En ce sens, les résultats fournis par une méthode d'évaluation quantitative sont donc tout à fait complémentaires de ceux obtenus à partir des outils conventionnels d'observation de la sécurité d'un système informatique.

3.2.1.7 Conclusion

En définitive, les résultats de cette expérience montrent que l'hypothèse MT est satisfaisante puisque le comportement de la mesure correspondante fournit des informations pertinentes à l'administrateur de sécurité pour ce qui est de la surveillance de l'évolution de la sécurité opérationnelle du système. Malheureusement, en raison de la complexité de l'algorithme utilisé, cette mesure n'a pu être obtenue en permanence. D'un autre côté, le calcul de la mesure correspondant à l'hypothèse ML est plus facile. Toutefois, il peut être plus difficile pour l'administrateur de sécurité d'identifier les actions à mettre en œuvre dans le système en se basant seulement sur cette mesure. En effet, dans ce cas, toute variation de la mesure doit être analysée, tandis que, dans le cas de l'hypothèse MT, on peut se contenter d'analyser les événements à l'origine d'une diminution de la mesure. Enfin, l'expérience montre que la longueur du plus court chemin, le nombre de vulnérabilités et le nombre de chemins ne sont pas suffisants pour caractériser l'évolution de la sécurité opérationnelle. Une analyse plus détaillée de ces différents résultats, et notamment de ceux mis en évidence dans les figures peut être trouvée dans [Ortalo *et al*. 1997].

Globalement, ces résultats expérimentaux confirment donc la faisabilité de l'approche d'évaluation quantitative de la sécurité, et la validité des différentes mesures proposées. Le prototype utilisé permet d'envisager une évolution vers un outil opérationnel, notamment dans le cadre d'un système informatique UNIX. Vis-à-vis de ce système informatique, un tel outil permet alors de surveiller l'évolution de la sécurité (dans le but de l'orienter) plutôt que d'éliminer systématiquement les vulnérabilités identifiées (qui peuvent être très nombreuses).

3.2.2 Application à une organisation

Dans le cadre d'une organisation, une observation automatique fréquente du système ne constitue pas une hypothèse de mise en œuvre réaliste. Néanmoins, la spécification de la politique de sécurité jointe à l'évaluation quantitative offre

l'opportunité d'étudier le fonctionnement réel de l'organisation et l'impact de modifications éventuelles de ce fonctionnement sur la sécurité. Afin d'illustrer les apports de cette approche, nous présentons à présent un tel exemple d'application, également utilisé dans [Ortalo & Deswarte 1998b].

L'organisation considérée est celle dont une spécification de la politique de sécurité a été présentée au 2.3.1.6, page 91. Il s'agit donc d'une agence bancaire de taille moyenne, comptant une trentaine de personnes. Étant donné la politique de sécurité considérée, qui reste partielle, nous nous intéressons à l'évaluation des objectifs de sécurité de haut niveau présentés dans la figure 18, page 96, vis-à-vis d'un petit nombre de vulnérabilités caractéristiques de l'étude d'un système d'information essentiellement appuyé sur des individus. L'objectif de cette application est de montrer que la méthode d'évaluation est utilisable dans le contexte d'une organisation, autorise la prise en compte de vulnérabilités très générales, et permet d'étudier l'impact de modifications éventuelles du fonctionnement sur la sécurité.

3.2.2.1 Vulnérabilités prises en compte

Afin d'étudier l'impact d'un certain nombre de vulnérabilités sur les objectifs de sécurité, la figure 31 présente deux vulnérabilités prises en compte dans l'organisation. Tout d'abord, on considère que les relations de confiance existant entre les différents agents de l'organisation peuvent permettre à un agent d'abuser des pouvoirs d'un autre agent. Nous nous intéresserons principalement à cette vulnérabilité. En effet, nous disposons d'un certain nombre d'informations concernant les relations de confiance existant effectivement dans l'organisation représentée, et nous pouvons donc nous intéresser à l'impact que cette confiance peut avoir sur le deuxième objectif de sécurité (figure 18, page 96).

Ensuite, nous considérons le cas où un agent appartenant à l'organisation peut profiter de la confiance que lui accorde un client pour effectuer des opérations à son détriment. Du point de vue de la modélisation des vulnérabilités, ceci revient à dire que si un agent possède la confiance d'un de ses clients, il lui est possible d'atteindre un état dans lequel aucune plainte ne provient de ce client malgré l'existence d'une malversation. Toutefois, cette vulnérabilité doit être enrichie pour représenter une situation réaliste. En effet, dans le domaine bancaire, on ne peut envisager raisonnablement qu'un agent ne réussisse réellement à abuser de la confiance de ses clients que dans des situations bien particulières. La plupart des opérations couramment effectuées font l'objet de la délivrance d'un reçu au client et un relevé complet de toutes les opérations courantes effectuées sur les comptes est régulièrement communiqué à leurs propriétaires. Ces documents font l'objet d'une vérification plus ou moins attentive par leur destinataire, toutefois étant donné qu'il s'agit de manipulation d'argent, on ne peut pas considérer que la confiance accordée par un client à son chargé de portefeuille suffise à le rendre totalement inattentif. De plus, le montant généralement faible des opérations courantes rend très peu intéressante l'exploitation de cette confiance en regard du risque encouru. Une telle vulnérabilité ne doit donc être considérée que dans le cadre de la mise en œuvre d'opérations

particulières qui se prêtent à une manipulation frauduleuse tirant parti de la confiance accordée. Ceci est notamment le cas pour les achats anonymes de titres au porteur. Dans ce type d'opération, qui fait d'ailleurs à l'heure actuelle l'objet d'une vérification extrêmement scrupuleuse de la part de tous les organismes bancaires pour éviter les infractions des clients eux-mêmes[1], les titres émis par la banque possèdent une valeur faciale importante. De plus, dans le cas des opérations anonymes, ils ne donnent pas lieu à l'émission de pièces comptables complètes. Dans ce contexte, l'exploitation de la confiance d'un client par l'agent qui lui a conseillé le placement est envisageable. C'est ce type de vulnérabilité qui est représenté dans la figure 31.

Vulnérabilités
Agents × Fait confiance × Autre agent ∧ **P**(Agents × Actions) ⇒ ◇ (**P**(Autre agent × Actions)) Client × Fait confiance × Agents ∧ Agents × Opérations anonymes sur titre ⇒ ◇ (¬(Client × Plainte))

Figure 31 - Vulnérabilités prises en compte

Cette liste de deux vulnérabilités n'est bien évidemment pas exhaustive, et on pourrait considérer d'autres vulnérabilités qui existent dans cette organisation. Par exemple, il est possible qu'un client accède à un terminal inoccupé pour tenter de faire des opérations à la place d'un des employés de la banque, ce qui peut avoir des conséquences plus ou moins importantes suivant le niveau d'habilitation de ce terminal, et peut être plus ou moins aisé du fait de la surveillance des employés, de la mise hors service automatique en cas d'inactivité, etc. Dans notre étude, nous nous sommes avant tout attaché à l'analyse de vulnérabilités d'assez haut niveau, caractéristiques d'une organisation, et ne nécessitant pas une description détaillée du fonctionnement de l'organisation.

Afin de réaliser l'évaluation quantitative, il est nécessaire d'affecter à ces différentes vulnérabilités un taux de succès correspondant à la difficulté de les mettre en œuvre. Nous considérons les deux vulnérabilités présentées dans la figure 31 dans les études présentées ci-après. Les taux de succès associés à ces deux vulnérabilités sont présentés dans le tableau 11, où λ_a est le taux de succès de l'attaque élémentaire consistant à exploiter la confiance d'un autre agent, et λ_c le taux de succès de l'attaque élémentaire consistant à exploiter la confiance d'un client. Compte tenu

1. Étant donné l'opportunité que ces produits bancaires offrent également aux opérations de dissimulation de revenus non-déclarés et surtout de blanchiment d'argent, la réglementation nationale concernant les opérations bancaires anonymes conduira d'ailleurs vraisemblablement à leur disparition à partir de 1998 (que ce soit pour l'émission ou pour la conversion des titres).

des valeurs choisies dans le tableau 11, on considère donc qu'il est deux fois plus difficile pour un agent de détourner à son profit la confiance d'un de ses clients que celle d'un de ses collègues.

Taux	Valeur
λ_a	1
λ_c	$1/2$

Tableau 11 - Quantification élémentaire

3.2.2.2 Construction du graphe des privilèges

La prise en compte de vulnérabilités dans le fonctionnement de l'organisation considérée conduit ensuite à reconsidérer ses différents objectifs de sécurité (figure 18, page 96) afin de savoir si ces différentes vulnérabilités peuvent permettre de les mettre en défaut. Dans le cas où c'est effectivement possible, la construction du graphe des privilèges correspondant est la première étape permettant d'aller vers une évaluation quantitative de l'impact de ces vulnérabilités sur la sécurité de l'organisation.

3.2.2.2.1 Premier cas

Nous nous intéressons tout d'abord à l'étude du premier objectif de sécurité présenté dans la figure 18, page 96, qui correspond au respect des délégations de crédits par les agents. Afin de déterminer les différents privilèges qui peuvent permettre d'enfreindre cet objectif, nous étudions donc la négation de cet objectif donnée par (40).

$$\begin{aligned}\mathbf{P}(\text{Agents} \times (\text{Accorder un crédit} \times \text{Crédits}) \wedge \text{Agents} \times \text{Emplois} \wedge \\ \neg\,\text{Autorisations de crédit} \wedge \text{Crédits} \times \neg(<500\text{kF}))\end{aligned} \tag{40}$$

Étant donnée la description du fonctionnement et notamment des autorisations de crédit, (40) correspond aux privilèges définis par (41), c'est-à-dire aux privilèges des agents occupant les emplois de responsable d'agence ou d'animateur qui sont les seuls à pouvoir accorder un crédit du montant indiqué.

$$\left\{\begin{array}{l}\mathbf{P}(\text{A1} \times (\text{Crédits} \times \neg(<500\text{kF}))) \\ \mathbf{P}(\text{A2} \times (\text{Crédits} \times \neg(<500\text{kF}))) \\ \mathbf{P}(\text{A5} \times (\text{Crédit à la consommation} \times <600\text{kF})) \\ \mathbf{P}(\text{A6} \times (\text{Crédit à la consommation} \times <600\text{kF})) \\ \mathbf{P}(\text{A7} \times (\text{Crédit à la consommation} \times <600\text{kF})) \\ \mathbf{P}(\text{A29} \times (\text{Crédit à la consommation} \times <600\text{kF}))\end{array}\right. \tag{41}$$

Étant donnée la vulnérabilité considérée, le seul moyen pour qu'un autre agent obtienne les privilèges de ces agents est d'abuser de la confiance des autres agents. C'est l'étude effectuée sur le terrain qui a permis d'identifier les cas où cette confiance est suffisamment forte pour que l'on puisse considérer que son exploitation

permet réellement d'effectuer un transfert de privilèges. Cette recherche a essentiellement consisté en une série d'entretiens effectués avec tout le personnel de l'agence bancaire. En considérant cette vulnérabilité, on obtient alors le graphe des privilèges présenté dans la figure 32[1], où chaque arc représente un lien de confiance fort entre les agents représentés par les nœuds cible et origine de l'arc. Dans ce graphe, tous les arcs correspondent au même type de vulnérabilité et sont associés à la même valeur de quantification élémentaire λ_a .

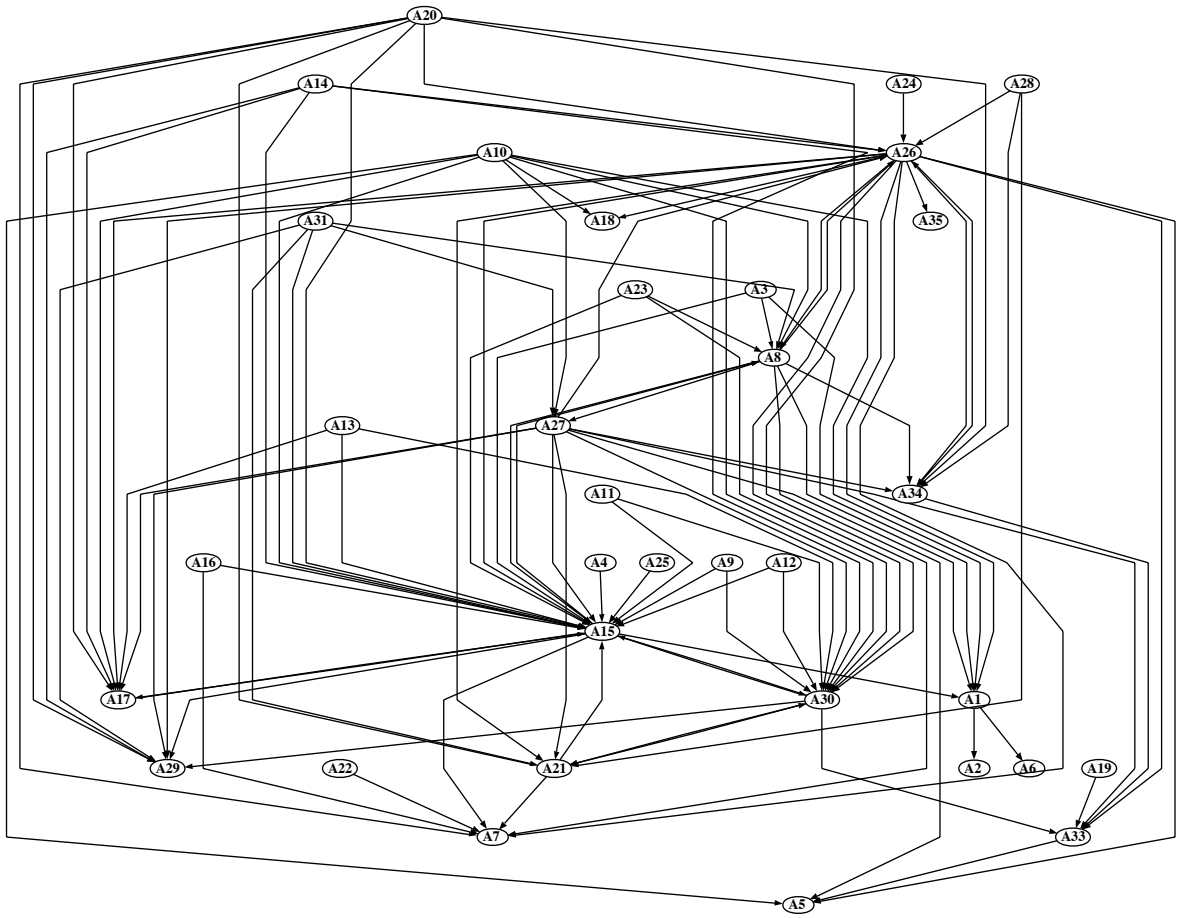


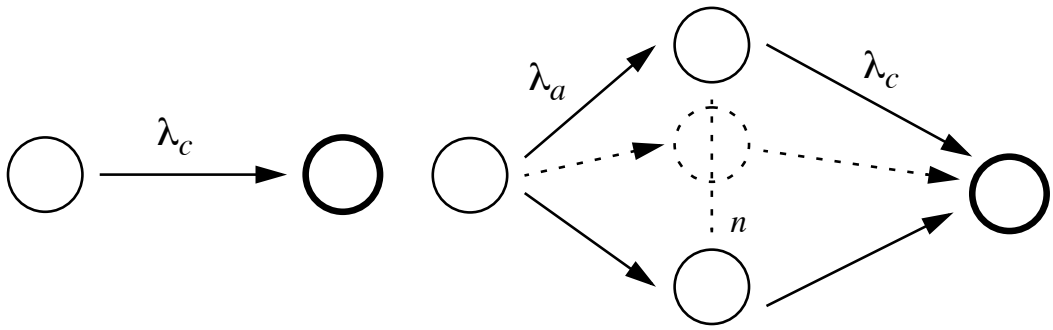
Figure 32 - Graphe des privilèges obtenu en considérant la première vulnérabilité (figure 31) et le premier objectif de sécurité (figure 18)[a]

a. Dans cette figure, on n'a porté aucun des arcs du graphe des privilèges dont l'origine partait d'une des *cibles* définies par l'objectif de sécurité.

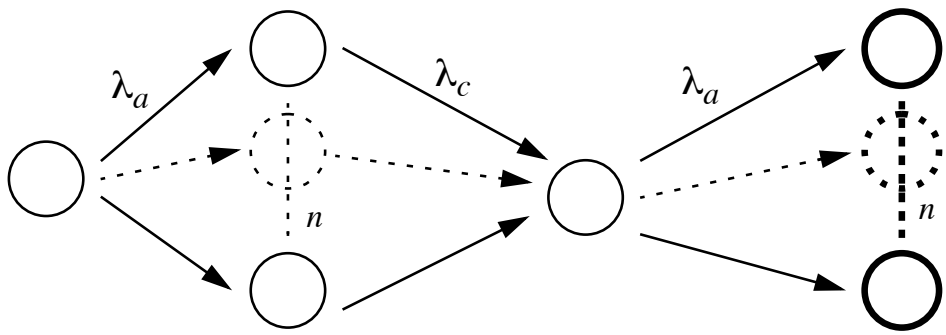
1. Cette figure a été construite automatiquement par l'outil de visualisation de graphes *daVinci* [Fröhlich & Werner 1994; Fröhlich & Werner 1996] à partir d'une description textuelle énumérant les différentes vulnérabilités identifiées. À la vue du graphe des privilèges, on peut remarquer que son exploitation directe serait très laborieuse, même dans le cas d'un nombre réduit de vulnérabilités.

3.2.2.2.2 Deuxième cas

Quand nous considérons la deuxième des vulnérabilités présentées dans la figure 31, le processus d'intrusion obtenu est extrêmement simple (figure 33-a) et ne contient que deux états qui correspondent tout simplement à l'exploitation de la vulnérabilité, laquelle permet de mettre en défaut le deuxième objectif de sécurité de l'organisation.



(a) Cas initial **(b)** Intégration d'une validation à l'émission des titres



(c) Intégration d'une validation à l'émission et à la conversion des titres

Figure 33 - Processus d'intrusion correspondant aux deux vulnérabilités (figure 31)

Face à ce type de vulnérabilité, on peut envisager des modifications du fonctionnement de l'organisation qui rendraient plus difficile son exploitation. Une solution relativement naturelle consiste à mettre en place un mécanisme de validation au moment de l'émission ou du rachat des titres. Dans ce cas, à chaque fois qu'un agent effectue ce type d'opération, la confirmation d'un deuxième agent doit être obtenue pour que l'opération soit effectivement effectuée. La mise en œuvre de ce type de fonctionnement semble possible, par exemple en associant des personnes différentes aux tâches de négociation commerciale et de délivrance proprement dite des titres. Dans ce cas, on doit donc modifier les règles de fonctionnement de l'organisation.

Dans cette hypothèse, la réalisation d'une manipulation frauduleuse nécessite donc l'exploitation de deux types de vulnérabilités : d'une part il s'agit de profiter de la confiance du client pour éviter qu'il ne suspecte le détournement de l'opération, mais il faut également exploiter la confiance des autres agents afin d'obtenir leur

validation soit à l'émission, soit au rachat, soit aux deux étapes du cycle de vie de ces titres. Dans ce cas, on obtient un processus d'intrusion du type de ceux présentés figure 33-b (dans le cas d'une validation à l'émission seulement) et figure 33-c (dans le cas d'une validation à l'émission et au rachat). Dans ces figures, λ_a est le taux de succès de l'attaque élémentaire consistant à exploiter la confiance d'un autre agent, λ_c est le taux de succès de l'attaque élémentaire consistant à exploiter la confiance d'un client, et *n* représente le nombre d'agents de l'organisation qui font suffisamment confiance à l'attaquant pour que celui-ci puisse envisager d'obtenir indûment leur validation.

3.2.2.3 Évaluation quantitative

Une fois le graphe des privilèges construit, le calcul des mesures de sécurité définies au 3.1.3 devient possible. Nous présentons donc à présent les résultats fournis par l'évaluation quantitative dans les deux cas correspondant aux deux objectifs de sécurité étudiés dans l'organisation.

3.2.2.3.1 Premier cas

À partir du graphe des privilèges présenté dans la figure 32 on obtient les résultats présentés dans le tableau 12. Ces résultats correspondent aux différentes mesures de sécurité présentées précédemment (cf 3.1.3) obtenues en considérant successivement que chacun des agents de l'organisation représente un attaquant potentiel (à l'exception des agents constituant les cibles à atteindre, qui correspondent aux cases grisées dans ce tableau). Compte-tenu des informations recueillies, certains agents ne peuvent pas exploiter les différentes vulnérabilités étudiées pour mettre en défaut les objectifs de sécurité, dans ce cas le nombre de chemins correspondant est 0 et les mesures de sécurité n'ont pas de sens (ce qui est marqué par un —).

Agent	$METF_{MT}$	$METF_{ML}$	$METF_{SP}$	Nombre de chemins
A1				
A2				
A3	0,477	0,763	1	261
A4	1,277	1,556	2	65
A5				
A6				
A7				
A8	0,291	0,733	1	195
A9	0,834	1,172	2	137
A10	0,285	0,785	1	671
A11	0,834	1,172	2	137
A12	0,834	1,172	2	137
A13	0,768	1,300	2	202
A14	0,398	0,938	2	343

Tableau 12 - Résultats de l'évaluation vis-à-vis du non respect des délégations de crédit pour des montants <500kF

Agent	$METF_{MT}$	$METF_{ML}$	$METF_{SP}$	Nombre de chemins
A15	0,277	0,556	1	65
A16	0,611	0,778	1	66
A17	1,277	1,556	2	65
A18	—	—	—	0
A19	2,000	2,000	2	1
A20	0,282	0,848	1	549
A21	0,514	0,787	1	79
A22	1,000	1,000	1	1
A23	0,624	1,026	2	332
A24	1,194	1,792	2	140
A25	1,277	1,556	2	65
A26	0,194	1,556	2	65
A27	0,273	0,863	1	272
A28	0,705	1,296	2	345
A29				
A30	0,476	0,789	1	72
A31	0,374	0,788	1	612
A32	—	—	—	0
A33	1,000	1,000	1	1
A34	0,888	1,308	2	126
A35	—	—	—	0

Tableau 12 - Résultats de l'évaluation vis-à-vis du non respect des délégations de crédit pour des montants <500kF (...)

La figure 34 propose une représentation graphique des résultats pour la mesure $METF_{MT}$. Dans cette figure, les différents agents ont également été ordonnés dans l'ordre croissant du résultat obtenu. Ainsi qu'on pouvait le penser, on observe une

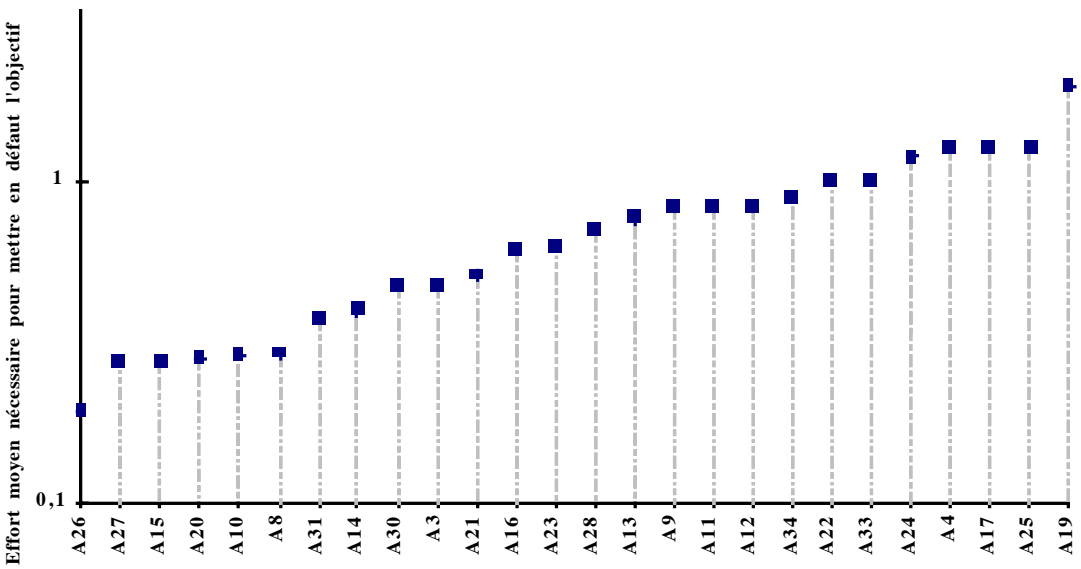


Figure 34 - Représentation graphique des résultats pour la mesure $METF_{MT}$ (respect des délégations de crédit <500kF)

répartition homogène des résultats numériques qui correspond aux opportunités plus ou moins nombreuses qu'un agent particulier peut avoir pour exploiter la confiance que lui portent les autres agents. Les agents auxquels on accorde le plus de confiance sont ceux qui sont susceptibles d'enfreindre le plus facilement l'objectif de sécurité considéré.

Ces résultats ne permettent pas directement d'agir sur le fonctionnement de l'organisation puisqu'il est assez illusoire et même probablement nuisible pour la vie quotidienne de l'agence d'envisager la réglementation des attitudes individuelles entre les membres du personnel. En revanche, la confrontation par un observateur extérieur de chaque résultat avec la situation de l'agent correspondant, du point de vue de son ancienneté ou de ses compétences par exemple, peut permettre de signaler des anomalies dans la répartition des responsabilités dans l'organisation. En effet, les liens de confiance professionnels qui existent entre les agents peuvent trouver leur origine dans des affinités individuelles, mais plus vraisemblablement, ils trouvent leur justification dans le travail quotidien et la manière dont celui-ci est mis en œuvre. L'identification de délégations tacites anormales de pouvoirs à des agents possédant peu d'ancienneté ou peu de compétences est par exemple certainement plus imputable à un cumul excessif de responsabilités qu'à des affinités individuelles. Cette charge peut être délicate à gérer au quotidien par l'agent qui accorde alors plus facilement sa confiance à un autre agent (en se basant forcément sur des critères subjectifs puisqu'aucune alternative ne lui est proposée par le fonctionnement de l'organisation).

Bien qu'une analyse puisse être envisagée, les résultats obtenus dans l'étude de cet objectif de sécurité particulier ne révèlent pas de tendances claires permettant d'identifier un dysfonctionnement. On notera toutefois qu'au sein de l'organisation considérée, les liens de confiance sont assez forts entre les différents agents.

Mais, l'objectif de ces calculs est avant tout d'illustrer la possibilité de mettre en œuvre la méthode d'évaluation quantitative de la sécurité dans le cadre d'une organisation. La prise en compte d'une vulnérabilité associée à la notion de confiance, qui apparaît comme une des notions parmi les plus délicates à utiliser dans l'extension de la méthode vers les systèmes d'information généraux, montre que l'on peut réellement envisager son application dans un cadre général. Les résultats obtenus sont cohérents avec le point de vue intuitif que l'on peut avoir au contact de l'organisation et permettent d'identifier les agents qui, de par leur position réelle dans l'organisation, sont les plus susceptibles de jouer sur des liens de confiance individuels. En revanche, aucune anomalie n'a été révélée. Ceci laisse supposer que les délégations implicites de pouvoirs dans cette organisation ne proviennent pas d'une mauvaise répartition des responsabilités.

3.2.2.3.2 Deuxième cas

Les expressions des mesures de sécurité correspondant aux trois processus d'intrusion présentés dans la figure 33 peuvent être obtenues directement en fonction de n, le nombre d'agents de l'organisation accordant leur confiance à l'agent malveillant. Ces expressions sont présentées dans le tableau 13.

$\mathrm{METF}_{(a)} = \frac{1}{\lambda_c}$	$\mathrm{METF}_{(b)} = \frac{1}{n\lambda_a} + \frac{1}{\lambda_c}$	$\mathrm{METF}_{(c)} = \frac{2}{n\lambda_a} + \frac{1}{\lambda_c}$
(a) Aucune validation	**(b)** Une validation	**(c)** Deux validations

Tableau 13 - Expressions comparées du METF

Comme on peut le remarquer, on a toujours $\mathrm{METF}_{(a)} < \mathrm{METF}_{(b)} < \mathrm{METF}_{(c)}$ ce qui correspond au fait que l'intégration d'une validation dans le fonctionnement de l'organisation au moment de l'émission ou du rachat d'un titre anonyme rend plus difficile la réalisation d'une manipulation frauduleuse par un agent de l'organisation. En effet, il est désormais nécessaire que l'agent malveillant exploite non seulement la confiance de son client, mais également celle d'un ou plusieurs de ses collègues afin de réussir sa manipulation. Toutefois, on note aussi que, étant donné les mesures considérées, cette valeur dépend de n, c'est à dire du nombre d'agents de l'organisation qui font confiance à l'agent malveillant. Plus nombreux sont les agents dont l'agent malveillant peut exploiter la confiance, et plus facile devient alors la mise en œuvre de l'opération (qui reste bien évidemment toujours plus difficile que d'exploiter seulement la confiance du client).

En considérant les valeurs numériques présentées dans le tableau 11 pour les taux de succès des deux vulnérabilités mises en jeu, on obtient, pour quelques valeurs de n, les résultats présentés dans le tableau 14.

n	$\mathrm{METF}_{(a)}$	$\mathrm{METF}_{(b)}$	$\mathrm{METF}_{(c)}$
1	2	3	4
2		2,5	3
3		2,333	2,666
4		2,25	2,5
5		2,2	2,4
6		2,166	2,333

Tableau 14 - Valeurs des mesures pour quelques valeurs de n

Les valeurs présentées dans le tableau 14 montrent que le gain de sécurité apporté par la mise en place d'une validation n'est significatif que dans les cas où la valeur de n est assez faible. Ceci signifie que le nombre de personnes qu'un agent malveillant est susceptible d'abuser afin d'obtenir leur validation doit rester limité pour que le mécanisme de validation apporte une amélioration notable. En pratique, il

semble donc nécessaire de réserver le pouvoir de valider l'émission (ou le rachat) anonyme d'un titre à un nombre limité de personnes (une ou deux dans le cas d'une seule validation à l'émission, entre 1 et 4 dans le cas de deux validations à l'émission et au rachat). Une modification du fonctionnement que l'on pourrait proposer consisterait donc :

- à maintenir pour tous les agents actuellement habilités à le faire l'autorisation d'effectuer des opérations anonymes sur les titres au porteur ;
- à mettre en œuvre, après la discussion commerciale, une étape supplémentaire associée par exemple à la délivrance ou à la remise effective du titre impliquant un nouvel agent, différent du premier, dont la validation est indispensable pour que l'opération soit définitivement menée à bien.

Dans ce cas de figure, les agents autorisés à effectuer la délivrance ou la remise effective du titre au porteur doivent rester en nombre relativement limité, et ne doivent pas être autorisés à traiter directement avec des clients sur ce type d'opération afin que le principe de la validation soit respecté dans leur cas.

On voit donc à travers cet exemple que l'utilisation d'une méthode d'évaluation quantitative de la sécurité dans le cadre d'une organisation permet d'étudier l'impact d'éventuelles modifications de son fonctionnement sur la sécurité. Cette approche permet donc d'améliorer la sécurité de l'organisation. La surveillance fréquente de la sécurité est probablement moins nécessaire dans une organisation que dans un système informatique car son fonctionnement change moins rapidement. La mise en œuvre de l'évaluation peut donc être limitée aux moments où des changements importants du fonctionnement ont lieu, contrairement au cas d'un système informatique dont l'observation fréquente semble nécessaire, ainsi que nous l'avons vu au 3.2.1.

3.2.2.4 Conclusion

Dans le cas étudié, le calcul des mesures de sécurité a pour objectif de montrer que la méthode d'évaluation quantitative peut être appliquée à une organisation très générale et que les résultats obtenus sont en conformité avec l'appréciation naturelle de la sécurité de l'organisation. Ces résultats permettent également d'étudier des évolutions éventuelles du fonctionnement de l'organisation destinées à améliorer sa sécurité, et de comparer l'impact sur la sécurité de plusieurs améliorations possibles. Étant donné que la spécification de la politique de sécurité de l'organisation est restée à un niveau d'abstraction assez élevé, et que l'on considère assez peu de vulnérabilités, les enseignements pratiques d'une telle étude restent toutefois limités. Dans le cadre d'une application réelle, il faudrait bien évidemment construire une description beaucoup plus détaillée de la politique de sécurité de l'organisation (notamment en ce qui concerne la description de son fonctionnement et de ses différentes règles de sécurité) et s'attacher à considérer un nombre plus important de vulnérabilités pour montrer l'intérêt pratique de l'utilisation d'une mesure de sécurité globale dans l'analyse de l'impact des différentes vulnérabilités.

Conclusion générale

Bilan

Dans ce mémoire, nous avons présenté une méthode permettant de mener à bien la spécification et l'évaluation quantitative de la sécurité d'un système d'information. Notre démarche a été la suivante :

- L'étude de la vision classique des politiques et modèles de sécurité effectuée au chapitre 1 a montré les difficultés d'application de ces approches à certains systèmes d'information. La plupart du temps, ces approches supposent en effet d'imposer des mécanismes ou des règles de fonctionnement dans le système, et suivant le contexte, induisent une rigidité parfois incompatible avec les autres exigences des utilisateurs. De même, les méthodes traditionnelles d'évaluation de la sécurité que nous avons présentées correspondent essentiellement à une vision statique de la sécurité, alors que l'étude de son évolution opérationnelle apparaît tout aussi importante en pratique.
- Pour résoudre ces problèmes, nous avons développé au chapitre 2 une méthode de spécification formelle d'une politique de sécurité susceptible de s'adapter au maximum à la réalité du fonctionnement du système d'information tout en permettant d'exprimer rigoureusement ses besoins de sécurité. L'utilisation de la logique déontique fournit une base formelle éprouvée pour l'étude de telles spécifications de la sécurité. Convenablement délimité, un langage logique structuré permet également de représenter assez facilement le fonctionnement et les besoins de sécurité d'une organisation, et semble même susceptible de fournir un cadre formel adéquat pour la représentation de ses vulnérabilités.
- Enfin, étant donné le point de vue adopté dans nos travaux en faveur de la prise en compte des vulnérabilités résiduelles existant dans le système, nous avons utilisé une méthode d'évaluation quantitative de la sécurité. Cette méthode, décrite au chapitre 3, fournit un moyen de juger de l'impact des vulnérabilités sur la sécurité du système. Elle peut donc permettre d'éviter l'élimination systématique de ces vulnérabilités, parfois irréaliste compte tenu du système d'information considéré. Nous avons également montré comment la spécification formelle de la politique de sécurité peut servir de point de départ à l'évaluation quantitative, afin de compléter l'ensemble de la méthode.

Cette démarche a été soutenue par une application expérimentale constante tout au long de son développement. On retiendra plus particulièrement que l'exploitation des résultats de la méthode d'évaluation quantitative, appliquée à un système informatique de grande taille pendant une longue période, a montré sa capacité à sur-

veiller l'évolution de la sécurité opérationnelle de ce système, et à fournir des informations correspondant aux besoins réels des administrateurs. Par ailleurs, l'application de la méthode de spécification à la définition de la politique de sécurité d'une organisation réelle illustre sa faisabilité, l'intérêt d'utiliser une décomposition structurée raffinée progressivement pour traiter ce genre de problème, et l'intérêt d'utiliser les opérateurs de la logique déontique pour formuler les objectifs de sécurité.

En conclusion, nous avons montré qu'il est possible et utile de définir une méthode générale d'évaluation de la sécurité, incluant une méthode de spécification des besoins de sécurité et une méthode d'évaluation quantitative permettant de surveiller son évolution opérationnelle. Adoptant à la fois un langage formel et une technique d'évaluation inspirée des mesures de sûreté de fonctionnement, la méthode présentée dans ce mémoire correspond à cet objectif et, nous l'espérons, démontre l'intérêt du point de vue que nous avons adopté.

Perspectives

Les choix effectués dans la définition de la méthode développée tout au long de ce mémoire ont essentiellement été guidés par le souci de définir complètement l'approche, de bout en bout. Un certain nombre de développements éventuels restent donc à étudier, et permettent de distinguer plusieurs perspectives de recherche ou de réalisation qui restent à explorer:

- L'application de la méthode de spécification d'une politique de sécurité à une organisation est restée à un haut niveau d'abstraction. Une étude détaillée consacrée à un système d'information complexe incluant de nouveaux aspects et notamment des procédures de fonctionnement plus différenciées devrait permettre d'identifier les problèmes les plus difficiles à résoudre dans le cadre formel proposé. Les systèmes d'information du secteur de la santé, et notamment les systèmes d'information hospitaliers, constitueraient un cas d'étude potentiel pour lequel un besoin se manifeste actuellement [Anderson 1996]. Cette étude permettrait alors de compléter le formalisme que nous avons défini, et d'étendre les outils logiciels permettant de l'utiliser.
- Le problème de l'exploitation pratique de la spécification formelle de la politique de sécurité dans l'objectif de sa vérification a seulement été évoqué dans ce mémoire, bien que plusieurs techniques de preuve aient été mentionnées. Le choix d'une méthode de vérification permettant d'exploiter de la manière la plus commode possible la spécification de la politique de sécurité, et la mise au point d'un outil de preuve adapté restent à étudier. L'exploitation de la spécification formelle permettrait alors d'améliorer dès sa conception le fonctionnement du système étudié et de ses mécanismes de sécurité.

- Les mesures quantitatives de la sécurité qui ont été présentées au chapitre 3 restent assez spécifiques. Il semble possible d'utiliser d'autres mesures, en s'inspirant des quelques travaux existant consacrés à l'évaluation quantitative de la sécurité [Moskowitz & Kang 1997; Reiter & Stubblebine 1997]. La définition et la mise en œuvre pratique de nouvelles mesures (appuyées sur les données déjà collectées dans les expériences que nous avons effectuées) permettrait d'effectuer une étude comparative des différents moyens d'obtenir une évaluation quantitative de la sécurité. L'objectif de ces extensions serait d'identifier précisément la mesure permettant de fournir à l'administrateur de sécurité du système les informations les plus pertinentes pour déterminer les modifications à proposer en vue d'améliorer la sécurité de ce système pendant sa vie opérationnelle.
- L'outil d'analyse d'un système informatique UNIX sur lequel repose les résultats expérimentaux présentés au chapitre 3.2.1 devrait être encore enrichi pour tenir compte de la plupart des vulnérabilités connues. Les aspects distribués étant d'ores et déjà pris en compte dans la version la plus récente de cet outil, la mise au point de ce dernier volet pourrait donner lieu à une mise en œuvre effective dans un environnement d'exploitation réel. Outre les enseignements théoriques qu'apporterait son utilisation, ceci constituerait la dernière étape préalable à sa diffusion ou sa commercialisation. Cette diffusion paraît souhaitable puisque, à notre connaissance, aucun autre outil de ce type n'existe à l'heure actuelle.
- Enfin, on peut se demander s'il est envisageable de construire une modélisation de la sécurité d'un système à partir d'une observation de son fonctionnement réel. Dans ce cas, on s'intéresse non plus aux objectifs de sécurité de ce système, c'est-à-dire à des propriétés attendues, mais aux propriétés effectivement assurées par le système, l'objectif étant de vérifier si elles correspondent effectivement aux besoins de sécurité au travers d'une synthèse de la politique de sécurité.

Annexe A Introduction à la logique modale

Nous présentons dans cette annexe les définitions fondamentales relatives à une logique modale. Nous détaillons également dans ce cadre classique quelques uns des axiomes les plus fréquemment étudiés qui sont à l'origine des différents systèmes de logique modale usuels. Enfin, nous nous tournons vers une présentation des logiques multimodales, c'est-à-dire des systèmes de logique modale faisant intervenir plusieurs opérateurs modaux simultanément. C'est dans ce cadre, plus général, que nous présentons certains résultats correspondant à la détermination et la décidabilité des systèmes de logique modale disponibles dans la littérature. En effet, ces conclusions sont importantes pour justifier du choix de la logique modale comme cadre formel pour la représentation des politiques de sécurité.

Les différents éléments mathématiques présentés ici sont extraits de [Chellas 1980] et [Catach 1989]. Des références plus précises sont données au fur et à mesure de la présentation des résultats.

A.1 Syntaxe

Cette section rappelle les concepts syntaxiques fondamentaux du langage de la logique modale (déjà présentés au 2.2.3.1, page 66). Le langage est fondé sur un ensemble dénombrable Φ de **propositions atomiques** $a, b, c, \ldots$ Celles-ci correspondent aux propositions les plus simples du langage.

Les propositions non atomiques sont construites au moyen de neuf opérations syntaxiques, ou **opérateurs**:

$$\top, \bot, \neg, \vee, \wedge, \Rightarrow, \Leftrightarrow, \Box, \Diamond$$

$\top$ et $\bot$ sont des opérateurs d'arité zéro, c'est-à-dire des constantes. $\top$ est la constante **vraie**, et $\bot$ est la constante **fausse**. $\neg$, $\Box$, et $\Diamond$ sont des opérateurs unaires; et $\vee$, $\wedge$, $\Rightarrow$, et $\Leftrightarrow$ sont des opérateurs binaires.

L'ensemble des propositions, ou **formules**, qui peut être construit est défini par:

(1) si $a \in \Phi$, a est une formule;
(2) $\top$ est une formule;
(3) $\bot$ est une formule;
(4) $\neg a$ est une formule ssi[1] a est une formule;
(5) $a \wedge b$ est une formule ssi a et b sont des formules;
(6) $a \vee b$ est une formule ssi a et b sont des formules;
(7) $a \Rightarrow b$ est une formule ssi a et b sont des formules;
(8) $a \Leftrightarrow b$ est une formule ssi a et b sont des formules;
(9) $\Box a$ est une formule ssi a est une formule;
(10) $\Diamond a$ est une formule ssi a est une formule.

Le langage ainsi défini est noté $L_{\Box}(\Phi)$. Les opérateurs $\Box$ et $\Diamond$ représentent respectivement les notions de *nécessité* et de *possibilité*. La traduction en langage naturel des formules $\Box p$ et $\Diamond p$ est donc respectivement "*Il est nécessaire que p*"[2] et "*Il est possible que p*".

Une **sous-formule** d'une formule f correspond à n'importe quelle formule qui est une partie de f. Cette notion intuitive peut être facilement capturée par une définition récursive simple [Chellas 1980, Déf. 2.2].

A.2 Modèles standards

Un **modèle standard** est une structure $\mathcal{M} = \langle W, R, V \rangle$ où W est un ensemble de mondes possibles, V représente une fonction associant à chaque proposition atomique un ensemble de mondes possibles, et R est une relation binaire entre les mondes possibles.

On désigne par $\models_w^{\mathcal{M}} f$ le fait que la formule f soit vraie dans le monde possible $w \in W$ dans le modèle $\mathcal{M}$. Cette valeur de vérité est définie par:

(1) $\models_w^{\mathcal{M}} a$, où $a \in \Phi$, ssi $w \in V(a)$

(2) on a $\models_w^{\mathcal{M}} \top$

(3) on n'a pas $\models_w^{\mathcal{M}} \bot$

(4) $\models_w^{\mathcal{M}} \neg f$ ssi on n'a pas $\models_w^{\mathcal{M}} f$

(5) $\models_w^{\mathcal{M}} f \wedge f'$ ssi $\models_w^{\mathcal{M}} f$ et $\models_w^{\mathcal{M}} f'$

1. "si et seulement si"
2. ou encore "*Nécessairement p*"

(6) $\models_w^{\mathcal{M}} f \vee f'$ ssi $\models_w^{\mathcal{M}} f$ ou $\models_w^{\mathcal{M}} f'$

(7) $\models_w^{\mathcal{M}} f \Rightarrow f'$ ssi, si $\models_w^{\mathcal{M}} f$ alors $\models_w^{\mathcal{M}} f'$

(8) $\models_w^{\mathcal{M}} f \Leftrightarrow f'$ ssi, $\models_w^{\mathcal{M}} f$ si et seulement si $\models_w^{\mathcal{M}} f'$

(9) $\models_w^{\mathcal{M}} \Box f$ ssi pour tout $w' \in W$ tel que wRw', $\models_{w'}^{\mathcal{M}} f$

(10) $\models_w^{\mathcal{M}} \Diamond f$ ssi il existe $w' \in W$ tel que wRw', $\models_{w'}^{\mathcal{M}} f$

On note par $\models^{\mathcal{M}} f$ le fait que pour tout $w \in W$, on ait $\models_w^{\mathcal{M}} f$, c'est-à-dire le fait que f soit vraie dans tous les mondes du modèle $\mathcal{M}$, ce qui est abrégé en disant que f est ***vraie*** *dans le modèle* $\mathcal{M}$. Enfin, on dit que f est ***valide*** *dans une classe de modèles* $\boldsymbol{\mathcal{C}}$, noté $\models_{\mathcal{C}} f$, si et seulement si elle est vraie dans tous les modèles de cette classe, c'est-à-dire vraie dans tous les mondes de tous les modèles de la classe.

Si $\boldsymbol{\mathcal{C}}$ est une classe de modèles *standards*, alors on a [Chellas 1980, Th. 3.3]:

DF $\models_{\mathcal{C}} \Diamond f \Leftrightarrow \neg \Box \neg f$

(2) $\forall (f_1, \ldots, f_n) \in \Phi^n$, si $\models_{\mathcal{C}} f_1 \wedge \ldots \wedge f_n \Rightarrow f$ alors

$\models_{\mathcal{C}} \Box f_1 \wedge \ldots \wedge \Box f_n \Rightarrow \Box f$

Ces propriétés caractérisent un système de logique modale **normal**, défini comme incluant le schéma d'axiome indiqué ci-dessus par DF et la règle d'inférence présentée par (2) qui correspond à RK (page 68) [Chellas 1980, Déf. 4.1].

Un système normal de logique modale possède également la règle d'inférence et le théorème suivants [Chellas 1980, Th. 4.2]:

$$\frac{f}{\Box f} \qquad \text{RN}$$

$$\Box (f \Rightarrow f') \Rightarrow (\Box f \Rightarrow \Box f') \qquad \text{K}$$

Et ce résultat permet de proposer une nouvelle axiomatisation d'un système normal. En effet, un système de logique modale contenant DF est normal ssi il contient K et est fermé[1] pour RN. D'autres axiomatisations alternatives peuvent également être considérées [Chellas 1980, Th. 4.3]. Étant donné ce résultat qui montre le rôle joué par l'axiome K dans une logique modale normale, le plus petit système de logique modale normal est également communément désigné par la lettre K.

1. cf note 1, page 67

A.3 Schémas d'axiomes

A présent, considérons des extensions du système normal K obtenues en lui ajoutant les axiomes suivants:

$$\Box f \Rightarrow \Diamond f \quad \text{D}$$

$$\Box f \Rightarrow f \quad \text{T}$$

$$f \Rightarrow \Box \Diamond f \quad \text{B}$$

$$\Box f \Rightarrow \Box \Box f \quad 4$$

$$\Diamond f \Rightarrow \Box \Diamond f \quad 5$$

On peut construire quinze systèmes normaux à partir d'une association quelconque de K et de ces différents axiomes. Ces différents systèmes sont notés en fonction de cette construction. On parle ainsi des systèmes KD, KT, KT5, etc. Historiquement, les plus importants de ces systèmes sont KD, KT, KTB, KT4 et KT5. Les deux premiers sont généralement associés respectivement à la logique déontique et la logique aléthique, et sont parfois notés simplement D et T. Les trois autres systèmes — KTB, KT4, et KT5 — constituent respectivement le système *Brouwersche* (parfois appelé B) et les systèmes de Lewis S4 et S5.

Si on met en rapport ces différents axiomes avec la classe de modèles dans laquelle ils sont respectivement valides, on peut établir une équivalence entre ces axiomes et des propriétés générales de la relation R pour les modèles de cette classe [Chellas 1980, Th. 3.5]. Le tableau 15 présente ces différents résultats.

Axiome	Propriété de R	
D	sérielle	$\forall u \exists v / uRv$
T	réflexive	$\forall u, uRu$
B	symétrique	$\forall u \forall v, uRv \Rightarrow vRu$
4	transitive	$\forall u \forall v \forall w, uRv \wedge vRw \Rightarrow uRw$
5	euclidienne	$\forall u \forall v \forall w, uRv \wedge uRw \Rightarrow vRw$

Tableau 15 - Correspondance entre les axiomes considérés et les propriétés de R

A.4 Les logiques multimodales

Comme nous l'avons indiqué (cf 2.2.2, page 63) il existe plusieurs ordres de modalité (ontique, temporelle, déontique, épistémique, dynamique,…) ayant conduit aux diverses théories modales et à l'étude de multiples systèmes modaux. Pourtant, les développements de ces diverses théories se sont faits de manière relativement indépendante, chaque système modal restant généralement étudié séparément. Pourtant, la combinaison des modalités semble parfaitement naturelle si l'on se réfère à leur

emploi dans le langage naturel. Par exemple, la phrase "*X ne sait pas qu'il est déjà obligatoire que p*" fait appel simultanément aux modalités épistémiques, temporelles et déontiques. Dans le cadre de l'application du formalisme modal à la spécification de la sécurité, ce type de formulation offre d'ailleurs des opportunités très intéressantes, comme on a pu le voir pour la représentation d'une vulnérabilité (cf 2.2.4.4.2, page 78). De manière plus générale, on peut même dire que toute utilisation pratique des logiques modales requiert presque immédiatement l'utilisation simultanée de modalités multiples. Nous présentons à présent un certain nombre de résultats correspondant à l'étude des logiques multimodales qui généralisent ceux obtenus pour chaque système modal étudié individuellement. Un traitement complet particulièrement intéressant[1] des logiques multimodales dans leur ensemble pourra être trouvé dans [Catach 1989], d'où tous les résultats que nous présentons à présent sont extraits.

A.4.1 Syntaxe

Un langage multimodal est basé sur un ensemble OPS d'opérateurs logiques, dits opérateurs modaux. La définition du langage multimodal correspond à celle donnée précédemment (cf A.1) à condition de substituer aux deux dernières règles de construction syntaxique la règle suivante:

(1) si $O \in \mathrm{OPS}$ est un opérateur modal *n*-aire, $O(f_1, \ldots, f_n)$ est une formule ssi $f_1, \ldots, f_n$ sont des formules.

Le langage ainsi défini est noté $\mathrm{L}(\Phi, \mathrm{OPS})$[2]. L'intérêt de cette définition est de munir l'ensemble OPS d'un certain nombre d'opérations formelles en s'intéressant à une structure du type $\langle \mathrm{OPS}, \Theta \rangle$ ou Θ est un ensemble d'opérations θ sur OPS. On ajoute alors la règle suivante pour la construction du langage:

(2) si $\theta \in \Theta$ est d'arité *n* et si $O_1, \ldots, O_n$ sont des opérateurs modaux, alors $\theta(O_1, \ldots, O_n)$ est un opérateur modal.

Ces opérations θ formalisent ce que l'on peut appeler un calcul sur les opérateurs modaux. Le cas le plus favorable survient lorsque certaines de ces opérations sont définissables syntaxiquement, c'est-à-dire au niveau du langage et non par référence à une théorie logique. On dispose de certaines opérations directement définissables syntaxiquement:

- les opérations booléennes $\neg O$, $O \wedge O'$, $O \vee O'$ sur les opérateurs;
- le **dual** O^d d'un opérateur O: $O^d f = \neg O \neg f$;
- la **composée** $(O_1;\ldots;O_n)f = O_1 \ldots O_n f$;
- les itérations O^n d'un opérateur O pour $n \geq 0$.

---

1. Et unique à notre connaissance.
2. Donc: $\mathrm{L}_{\Box}(\Phi) = \mathrm{L}(\Phi, \{\Box, \Diamond\})$

Ces opérations constituent les opérations de base pour les langages multimodaux. L'opération de composition est une opération fondamentale qui permet notamment de formaliser la notion de contextes imbriqués.

Le calcul direct sur les opérateurs modaux présente certains avantages [Catach 1989, §2.8.1]. Néanmoins, d'un point de vue technique, il est plus commode de disposer d'une représentation uniforme des opérateurs modaux. Or, il se trouve que la logique dynamique propose déjà ce type de représentation avec des opérateurs notés:

$$[a], [b], \ldots \qquad \text{et} \qquad \langle a\rangle, \langle b\rangle, \ldots$$

où a et b représentent des programmes. En réalité, il est facile de voir que tout système multimodal peut être réinterprété dans une logique dynamique, à condition de considérer simplement $a, b, \ldots$ comme des paramètres formels, appelés **paramètres modaux**. On fait alors correspondre, à tout couple $(\Box, \Diamond)$ d'opérateurs *duaux* un paramètre a tel que $\Box = [a]$ et $\Diamond = \langle a\rangle$.

On désigne alors par Σ l'ensemble des paramètres modaux, et on note $L(\Sigma)$ le langage multimodal construit à partir de Σ. On dit qu'un paramètre a est n-aire si l'opérateur associé $[a]$ est un opérateur n-aire. Si a est unaire, on écrit $[a]f$ pour $[a](f)$, et si a est binaire, on écrit $f[a]f'$ au lieu de $[a](f, f')$. On note $\langle a\rangle$ l'opérateur dual de $[a]$, soit $\langle a\rangle(f_1, \ldots, f_n) = \neg[a](\neg f_1, \ldots, \neg f_n)$. L'ensemble OPS des opérateurs modaux du langage est donc alors:

$$\text{OPS} = \{[a]/a \in \Sigma\} \cup \{\langle a\rangle/a \in \Sigma\}$$

La formalisation du langage construit à partir d'un ensemble de paramètres modaux, notamment la notion de base atomique Σ_0 de paramètres et un certain nombre d'exemples, sont décrits plus précisément dans [Catach 1989, §2.8.3-2.8.15]. On mentionnera plus particulièrement l'introduction de la notion de composition des paramètres, notée $[a;b]$ ou $\langle a;b\rangle$, qui correspond à la composition des paramètres modaux: $[a;b]f = ([a];[b])f = [a][b]f$.

A.4.2 Schémas généraux d'axiomes

Nous avons vu précédemment certains axiomes particuliers étudiés sur les opérateurs modaux (cf A.3). La notation paramétrée permet de définir des schémas *généraux* d'axiomes. Les axiomes les plus populaires de la logique modale sont les axiomes D, T, B, 4 et 5. Une généralisation de ces axiomes est le schéma $G^{k, l, m, n}$ [Chellas 1980, §5.5; Catach 1989, §5.2.1] suivant:

$$G^{k, l, m, n} = \Diamond^k \Box^l f \Rightarrow \Box^m \Diamond^n f \tag{42}$$

Par exemple, les axiomes D, T, B, 4 et 5 correspondent respectivement à $G^{0, 1, 0, 1}$, $G^{0, 1, 0, 0}$, $G^{0, 0, 1, 1}$, $G^{0, 1, 2, 0}$ et $G^{1, 0, 1, 1}$.

On désigne par $G(a, b, c, d)$ le schéma d'axiome:

$$G(a, b, c, d) = \langle a\rangle[b]f \Rightarrow [c]\langle d\rangle f \tag{43}$$

L'axiome G(a, b, c, d) est introduit dans [Catach 1989, §5.2.2]. Le point important relatif à la généralité de cet axiome est que les paramètres a, b, c, d ne sont pas nécessairement atomiques, mais au contraire peuvent être des expressions complexes. Par exemple, il est clair que G(a, b, c, d) généralise $G^{k,l,m,n}$: si $\Box = [A]$ et $\Diamond = \langle A \rangle$, il suffit de prendre $a = A^k$, $b = A^l$, $c = A^m$, et $d = A^n$ où $A \in \Sigma_0$ est un paramètre atomique [Catach 1989, §2.8.3]. Par conséquent, G(a, b, c, d) couvre les axiomes D, T, B, 4 et 5 de la logique modale usuelle, et la classe des systèmes multimodaux construits à partir d'axiomes G(a, b, c, d) couvre les systèmes où chaque opérateur modal (atomique) appartient à l'un des 15 K-systèmes normaux usuels de la logique modale engendrés par D, T, B, 4 et 5. On peut remarquer que G(a, b, c, d) couvre également certains **axiomes d'interaction** entre diverses modalités. Si on prend les opérateurs modaux $\Box_1$ et $\Box_2$ représentés par des paramètres atomiques A et B ($\Box_1 = [A]$ et $\Box_2 = [B]$) on a en effet par exemple:

$K_{1,2}$	$\Box_2 f \Rightarrow \Box_1 f$	G(λ, B, A, λ)
$D_{1,2}$	$\Box_2 f \Rightarrow \Diamond_1 f$	G(λ, B, λ, A)
$B_{1,2}$	$f \Rightarrow \Box_1 \Diamond_2 f$	G(λ, λ, A, B)
$4_{1,2}$	$\Box_1 f \Rightarrow \Box_2 \Box_1 f$	G(λ, A, B;A, λ)
$5_{1,2}$	$\Diamond_1 f \Rightarrow \Box_2 \Diamond_1 f$	G(A, λ, B, A)
$SC_{1,2}$	$\Box_2 \Box_1 f \Rightarrow \Box_1 \Box_2 f$	G(λ, B;A, A;B, λ)

De très nombreux principes d'interaction que l'on peut envisager sont en réalité du type G(a, b, c, d) [Catach 1989, §5.2.2].

On peut encore étendre de manière notable la généralité de ces axiomes en considérant le schéma d'axiome G(a, b, φ) suivant introduit par [Catach 1989, §5.2.9]:

$$G(a, b, \varphi) = \langle a \rangle [b] f \Rightarrow \varphi \qquad (44)$$

Dans cette définition, comme précédemment a et b sont des paramètres modaux quelconques, et φ est une formule *affirmative*[1]. L'axiome G(a, b, c, d) est un cas particulier de G(a, b, φ) où $\varphi = [c]\langle d \rangle f$. Mais cet axiome est plus général, on trouvera un certain nombre d'exemples et d'illustrations de cet axiome, notamment dans les cas où il ne se ramène pas au schéma G(a, b, c, d), dans [Catach 1989, §5.2.10-5.2.11].

1. Informellement, f est une formule affirmative si elle correspond à un nombre pair de négations vis-à-vis de toutes les propositions dont elle dépend, l'application d'un opérateur modal ne modifiant pas cette polarité. Par exemple, $\Box p_1 \wedge p_2$ est affirmative, $\neg \Diamond p_1 \vee \neg p_2$ est négative, $\Box p_1 \wedge \neg p_2$ est affirmative en p_1 et négative en p_2, mais n'a *pas* de polarité. En particulier, on notera que $\Box f$ et $\Diamond f$ ont la même polarité. La définition formelle d'une formule affirmative pourra être trouvée dans [Catach 1989, §2.1.10].

A.4.3 Détermination

Le problème de la détermination (correction et complétude) d'un système logique L relativement à une sémantique S consiste à étudier quels sont les modèles de S validant exactement les théorèmes de L. D'une manière équivalente, la détermination étudie l'adéquation entre l'axiomatique et la sémantique d'un système logique; il s'agit donc d'une question essentielle dans son étude. L'extension des résultats obtenus dans les systèmes monomodaux vers les systèmes multimodaux est une condition importante pour leur utilisation pratique. La plupart des résultats de [Catach 1989, Chap. 7 et 8] indiquent que l'on peut cumuler les résultats obtenus indépendamment [Catach 1989, Th. §8.3.1]. Ainsi, la correction est préservée par "extension" des systèmes modaux vers les systèmes multimodaux. En particulier, comme on pouvait le supposer, seule la correction concernant les *axiomes d'interaction* peut éventuellement poser un problème lorsque l'on considère des systèmes multimodaux construits en superposant des systèmes monomodaux. On peut donc se restreindre à étudier la correction pour les axiomes d'interaction uniquement [Catach 1989, §8.3.2]. Par ailleurs, la complétude d'un système multimodal peut être montrée pour tous les systèmes normaux construits à partir des schémas d'axiomes G(a, b, φ) [Catach 1989, Th. §8.3.11]. Ceci conduit donc à un résultat général de détermination pour tous les systèmes multimodaux normaux construits à partir de ces schémas.

A.4.4 Décidabilité

Enfin, l'étude des systèmes multimodaux présentée dans cette section conduit à un certain nombre de résultats s'appliquant de manière générale à la fois aux systèmes multimodaux et aux systèmes monomodaux.

Notamment, si $L_1, \ldots, L_n$ sont des systèmes modaux pris dans la liste :

$$L_i = \{\text{K, KD, KB, K4, K45, KB4, KD4, KD45,}\\ \text{KDB, KT, KT4, KTB, KT5, KTr, KV}\}$$

alors le système multimodal $L = L_1 \times \ldots \times L_n$ est décidable [Catach 1989, Th. §8.4.15].

Notons que les systèmes multimodaux L considérés dans ce théorème sont des systèmes *sans interactions*, mais non nécessairement homogènes (car les sous-systèmes peuvent être de différents types).

Enfin, nous avons le résultat suivant concernant les systèmes avec interactions. Si $L_1, \ldots, L_n$ sont des systèmes modaux pris parmi les 15 systèmes de logique modale indiqués précédemment, si $\Sigma_0 = \{A_1, \ldots, A_n\}$, soient $S_1, \ldots, S_p$ des schémas d'axiomes du type:

$$[A_j]f \Rightarrow [A_i]f \qquad \text{ou} \qquad \langle A_i \rangle f \Rightarrow \langle A_j \rangle f$$

tels que, si (i, j) intervient dans un axiome de ce type, ou bien $L_i = \{K, KD, KB, KDB, KT, KTB, KTr, KV\}$ ou bien $L_j = \{K, KD, KT\}$, alors, le système de logique multimodale $L = L_1 \times \ldots \times L_n + S_1 \ldots S_p$ est décidable [Catach 1989, Th. §8.4.17].

Annexe B Analyse détaillée des événements de sécurité

Les descriptions détaillées des causes de chaque variation significative des mesures de sécurité apparaissant dans la figure 27 (page 134) et la figure 28 (page 135) sont présentées dans le tableau 16. Tous les événements de sécurité qui sont survenus pendant l'expérience présentée au 3.2.1 (page 129) ne sont pas mentionnés ici : seuls sont considérés ceux qui ont conduit à une variation significative d'une des mesures. Dans ce tableau, ΔNP, ΔML et ΔTM désignent respectivement : la variation absolue du nombre de chemins, et les variations relatives de $METF_{ML}$ et $METF_{TM}$. "—" signifie que la valeur n'est pas disponible, et "~0" signifie que la valeur absolue de la variation relative observée est inférieure à 0,5 %.

				Objectif 1			Objectif 2		
Date	#	Utilisateurs impliqués	Description	ΔNP	ΔML (%)	ΔTM (%)	ΔNP	ΔML (%)	ΔTM (%)
1995									
17 Juin	1	U1	Le mot de passe de U1 peut être trouvé.	+2	~0	~0	-1	~0	—
18 Juin	2	U2	Le répertoire principal de U2 devient inscriptible.	+2	-3	-13	+1	+1	—
19 Juin	3	U3, U4	Une attaque par cheval de Troie sur le fichier `.login` de U3 et le fichier `.tcshrc` de U4 devient moins probable en raison d'un changement de `shell`.	0	+9	+22	0	-2	—
20 Juin	4	U3, U4	(Opposé de #3.)	0	-8	-18	0	+2	—
29 Juin	5	U21	Le répertoire principal de U21 devient inscriptible.	0	0	0	+1	-7	—
24 Août	6	U12, et de nombreux autres	U12 active une vulnérabilité exploitable par près de six cents autres utilisateurs.	0	0	0	+3445	+108	—
7 Sept.	7	U5	Une attaque par cheval de Troie sur le fichier `.login` de U5 devient possible.	+2	-2	-8	+244	~0	—
9 Sept.	8	U13, U14	Le mot de passe de U13 ne peut plus être trouvé. U14 corrige le groupe de certains de ses fichiers d'initialisation. (Ceci élimine #6.)	0	0	0	-2867	-54	—
14 Sept.	9	U6	Le répertoire principal de U6 devient inscriptible.	+2	-2	-6	1	+3	-1
20 Sept.	10	U6	(Opposé de #9.)	-2	+2	+7	-2	-2	+1
18 Oct.	11	U7	Le fichier `.rhosts` de U7 devient inscriptible.	+2	-8	-27	+1	+13	-5
20 Oct.	12	U7	(Opposé de #11.)	-2	+8	+29	-1	-12	+6

Tableau 16 - Description détaillée des événements de sécurité

Date	#	Utilisateurs impliqués	Description	Objectif 1 ΔNP	Objectif 1 ΔML (%)	Objectif 1 ΔTM (%)	Objectif 2 ΔNP	Objectif 2 ΔML (%)	Objectif 2 ΔTM (%)
25 Oct.	13	U5	Une attaque par cheval de Troie concernant plusieurs fichiers d'initialisation de U5 devient possible pour les membres de son groupe.	+46	+74	—	+24	+70	—
29 Nov.	14	U5,U8	U5 retire U8 de son fichier `.rhosts`. (Ceci élimine #13.)	-46	-41	—	-24	-43	—
1 Déc.	15	U9	Une nouvelle attaque devient possible par l'intermédiaire d'un fichier système.	+5	-19	-9	0	0	0
19 Déc.	16	U15, U16, U17	Des attaques potentielles par cheval de Troie concernant U15 et U16 sont éliminées. Le mot de passe de U17 peut être trouvé.	0	0	0	+1	+8	+20
20 Déc.	17	U15, U16	Les attaques possibles sur U15 et U16 redeviennent exploitables (cf #8).	0	0	0	0	-8	-17
1996									
3 Janv.	18	U10	Le mot de passe de U10 peut être trouvé.	+3	~0	~0	1	~0	~0
5 Janv.	19	U1, U10, U15, U16, U17, U18, U19, U20	*Objectif 1*: Les mots de passe de U10 et U1 ne peuvent plus être trouvés. *Objectif 2*: Les mots de passe de U1, U10, U15, U16, U17, U18, U19, et U20 ne peuvent plus être trouvés. (Ceci corrige un problème identifié dans #16.)	-6	~0	~0	-6	+1	+2
13 Janv.	20	U1, U10, U15, U16, U17, U18, U19, U20	*Objectif 1*: Le mot de passe de U1 peut à nouveau être trouvé (cf #1 et #19). *Objectif 2*: Les mots de passe de U1, U10, U15, U16, U17, U18, U19, et U20 peuvent à nouveau être trouvés (cf #19 et # 16).	+3	~0	~0	+5	-1	-2
30 Janv.	21	`root`	`root` devient un membre du groupe `admin_group`.	0	0	0	+15	-63	—
1 Fév.	22	U11	Le fichier de configuration `.mailrc` de U11 devient inscriptible.	+3	~0	-1	+4	~0	—
2 Fév.	23	`root`	`root` n'est plus membre du groupe `admin_group`. (Opposé de #21.)	0	0	0	-18	+170	—
21 Mars	24	U2, U3, U11, U15, U16, U20, U22, U23, U25, U26, `root`	*Objectif 1*: Les attaques sur U2, U3 et U11 ne sont plus exploitables. (Opposé de #2, #3 et #22 respectivement.) Une nouvelle attaque est possible par l'intermédiaire d'un fichier du système pour U22. Le mot de passe de U23 peut être trouvé. Plusieurs attaques directes visant `root` deviennent exploitables pour n'importe qui. *Objectif 2*: Les attaques vers U15 et U16 sont à nouveau désactivées (cf #17). Les attaques sur U2, U3, U11 et U20 sont désactivées. U20 n'est plus une cible correspondant à l'objectif de sécurité. U23 acquiert plus de privilèges (il devient membre d'un nouveau groupe) et son mot de passe peut être trouvé. U25 devient vulnérable à une attaque par cheval de Troie effectuée par U26.	0	-98	-98	+57	+1173	—

Tableau 16 - Description détaillée des événements de sécurité (...)

Date	#	Utilisateurs impliqués	Description	Objectif 1			Objectif 2		
				ΔNP	ΔML (%)	ΔTM (%)	ΔNP	ΔML (%)	ΔTM (%)
22 Mars	25	U15, U16	Les attaques sur U15 et U16 sont à nouveau exploitables (cf #16, #17 et #24).	0	0	0	+120	-66	—
28 Mars	26	U27	Le mot de passe de U27 peut être trouvé.	+3	~0	—	+37	~0	—
3 Avril	27	U28	Le mot de passe de U28 peut être trouvé.	+3	~0	—	+37	~0	—
5 Avril	28	U32, U24, `root`	*Objectif 1*: Le répertoire principal de U24 devient inscriptible. Une attaque directe vers `root` est exploitable. *Objectif 2*: U32 n'est plus une cible. Le répertoire principal de U24 devient inscriptible. Une attaque directe vers `root` est exploitable.	+3	+13	~0	+30	+1	—
11 Avril	29	U15, U16	Les attaques sur U15 et U16 sont à nouveau désactivées (cf #16, #17, #24 et #25).	0	0	0	-194	+200	—
12 Avril	30	U1	Le mot de passe de U1 peut être trouvé.	+3	~0	~0	+12	~0	—
16 Avril	31	U33, U34	U33 et U34 sont de nouvelles cibles. U33 et U34 sont vulnérables à une attaque par cheval de Troie sur certains de leurs fichiers d'initialisation.	0	0	0	+2	-2	—
19 Avril	32	U6	Le répertoire principal de U6 devient inscriptible.	+3	+16	~0	+12	~0	—
10 Mai	33	U6	Le répertoire principal de U6 n'est plus inscriptible.	-3	-14	~0	-12	~0	—
15 Mai	34	U1, U35	*Objectif 1*: Le mot de passe de U1 ne peut plus être trouvé. *Objectif 2*: U35 n'est plus une cible. Les mots de passe de U1 et U35 ne peuvent plus être trouvés.	-3	~0	~0	-13	~0	—
30 Mai	35	`root`, U9, U29, U30, U22, U26	La plupart des attaques directes concernant `root` sont désactivées (cf #24). Aucun des utilisateurs U9, U22, U26, U29 ou U30 ne peut plus l'attaquer grâce à un cheval de Troie (rarement susceptible d'être activé).	-22	+292	+740	-88	-14	—
31 Mai	36	`root`, U9, U29, U30, U22, U26	(Opposé de #35.)	+22	-73	-87	+88	+17	—
3 Juin	37	`root`	Une partie des attaques directes concernant `root` sont désactivées (cf #24).	0	+196	+68	0	-4	—
13 Juin	38	`root`	Deux attaques directes concernant `root` sont désactivées (cf #24).	0	-26	-28	0	+1	—
16 Juin	39	U32	U32 est à nouveau une cible, et son répertoire principal est inscriptible (par les membres d'un groupe différent du sien).	0	0	0	+7	-1	—
19 Juin	40	`root`	Une attaque directe vers `root` est à nouveau exploitable.	0	+15	+16	0	~0	—
29 Juin	41	`root`, U26	Une partie des attaques directes vers `root` sont désactivées (cf #24). La plupart des attaques vers `root` exploitables par U26 sont désactivées.	0	+154	+129	0	-8	—

Tableau 16 - Description détaillée des événements de sécurité (...)

Date	#	Utilisateurs impliqués	Description	Objectif 1			Objectif 2		
				ΔNP	ΔML (%)	ΔTM (%)	ΔNP	ΔML (%)	ΔTM (%)
3 Juillet	42	U36, U37	U36 et U37 sont de nouvelles cibles. Leurs mots de passe peuvent être trouvés.	0	0	0	+2	~0	—
4 Juillet	43	U31	Le mot de passe de U31 peut être trouvé.	+3	~0	~0	+13	~0	—
9 Juillet	44	U31, U26	Le mot de passe de U31 ne peut plus être trouvé. Le répertoire principal de U23 devient inscriptible.	0	+13	~0	0	~0	—
22 Juillet	45	`root`	Une attaque directe vers `root` est désactivée (cf #24).	0	+59	+76	-1	-7	—

Tableau 16 - Description détaillée des événements de sécurité (...)

RÉFÉRENCES BIBLIOGRAPHIQUES

[Abecassis 1995] A.-F. Abecassis, *"De plus en plus malmené, le secret médical est-il en danger ?"*, Objectif Soins, no.31, pp.22-23, mars 1995.

[Aerospatiale 1997] Aerospatiale, *PROLE2: Démarche et questionnaire*, 83 p., Aerospatiale, RAP/PROLE/DOC/96.007, v.1.2, 31 octobre, 1997. (Diffusion restreinte.)

[Alchourròn 1993] C. E. Alchourròn, "Philosophical Foundations of Deontic Logic and the Logic of Defeasible Conditionals", in *Deontic Logic in Computer Science*, (J.-J. C. Meyer, R. J. Wieringa, Eds.), pp.43-84, ISBN 0-471-93743-6, Wiley, Chichester, England, 1993.

[Anderson 1996] R. J. Anderson, "A Security Policy Model for Clinical Information Systems", in *IEEE Symposium on Security and Privacy,* Oakland, California, May 6-8, pp.30-43, ISBN 0-8186-7417-2, IEEE Computer Society Press, 1996.

[Aristote 1992] Aristote, *Organon - Les premiers analytiques,* Bibliothèque des textes philosophiques, vol. 3/6, 334 p., ISBN 2-7116-0017-3, Librairie Philosophique J. Vrin, Paris, 1992.

[Aslam 1995] T. Aslam, *A Taxonomy of Security Faults in the Unix Operating System,* Master of Science, Purdue University, USA, August, 1995.

[Beckert & Posegga 1995] B. Beckert, J. Posegga, "leanTAP: Lean Tableau-based Deduction", *Journal of Automated Reasoning*, vol.15, no.3, pp.339-358, 1995.

[Beeri *et al*. 1987] C. Beeri, S. Naqvi, R. Ramakrishnan, O. Shmueli, S. Tsur, "Sets and Negation in a Logic Database Language (LDL1)", in *6th annual ACM symposium on Principles of Database Systems (PODS),* San Diego, California, USA, pp.21-37, ISBN 0-89791-223-3, ACM, 1987.

[Beeri *et al*. 1991] C. Beeri, S. Naqvi, O. Shmueli, S. Tsur, "Set Constructors in a Logic Database Language", *Journal of Logic Programming*, vol.10, no.3&4, pp.181-232, April/May, 1991.

[Bell & LaPadula 1975] D. E. Bell, L. J. LaPadula, *Secure Computer Systems: Unified Exposition and Multics Interpretation*, The MITRE Corporation, ESD-TR-73-306, 1975.

[Beth *et al*. 1994] T. Beth, M. Borcherding, B. Klein, "Valuation of Trust in Open Networks", in *Third European Symposium on Research in Computer Security (ESORICS 94),* (D. Gollman, Ed.), Brighton, United Kingdom, Lecture Notes in Computer Science, 875, pp.3-18, ISBN 3-540-58618-0, Springer-Verlag, 1994.

[Biba 1977] K. J. Biba, *Integrity Considerations for Secure Computer Systems*, MITRE Corporation, Technical Report, ESD-TR-76-372 & MTR-3153, 1977.

[Bibel & Eder 1993] W. Bibel, E. Eder, "Methods and Calculi for Deduction", in *Handbook of Logic in Artificial Intelligence and Logic Programming, Logical Foundations*, (D. M. Gabbay, C. J. Hogger, J. A. Robinson, Eds.), vol. 1/5, pp.67-182, ISBN 0-19-853745-X, Oxford Science Publications, 1993.

[Bieber & Cuppens 1992] P. Bieber, F. Cuppens, "A Logical View of Secure Dependencies", *Journal of Computer Security*, vol.1, no.1, pp.99-129, 1992.

[Bieber & Cuppens 1993] P. Bieber, F. Cuppens, "Expression of Confidentiality Policies with Deontic Logic", in *Deontic Logic in Computer Science*, (J.-J. C. Meyer, R. J. Wieringa, Eds.), pp.103-123, ISBN 0-471-93743-6, John Wiley & Sons, 1993.

[Brewer & Nash 1989] D. Brewer, M. Nash, "The Chinese Wall Security Policy", in *IEEE Symposium on Security and Privacy,* Oakland, California, May 1-3, pp.206-214, ISBN 0-8186-1939-2, IEEE Computer Society Press, 1989.

[Brown 1994] M. A. Brown, "A Logic of Comparative Obligation", in *Second International Workshop on Deontic Logic in Computer Science,* Amsterdam (The Netherlands), pp.37-55, 1994.

[Calas 1994] C. Calas, "Distributed File System over a Multilevel Secure Architecture Problems and Solutions", in *Third European Symposium on Research in Computer Security (ESORICS 94),* (D. Gollman, Ed.), Brighton, United Kingdom, Lecture Notes in Computer Science, pp.281-297, ISBN 3-540-58618-0, Springer-Verlag, 1994.

[Calas 1995] C. Calas, *Pour une Protection Efficace des Données et des Traitements dans les Systèmes Informatiques Répartis,* Thèse de doctorat, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (ENSAE), no.176, 264p., 19 décembre, 1995.

[Catach 1989] L. Catach, *Les Logiques Multimodales,* Thèse de doctorat, Université Pierre et Marie Curie (Paris 6), Paris, France, 312p., 1989.

[Catach 1991] L. Catach, "TABLEAUX: A General Theorem Prover for Modal Logics", *Journal of Automated Reasoning*, vol.7, pp.489-510, 1991.

[CC 1996a] CC, *Common Criteria for Information Technology Security Evaluation*, *Part1: Introduction and general model*, 60p., CCEB-96/011, version 1.0, 1996.

[CC 1996b] CC, *Common Criteria for Information Technology Security Evaluation*, *Part4: Predefined Protection Profiles*, 166p., CCEB-96/014, version 1.0, 1996.

[Chellas 1980] B. F. Chellas, *Modal Logic: An Introduction,* 295p., ISBN 0-521-29515-7, Cambridge University Press, 1980.

[Chen & Sandhu 1995] F. Chen, R. S. Sandhu, "Constraints for Role-Based Access Control", in *1st ACM Workshop on Role-Based Access Control,* NIST, Gaithersburg, Maryland, USA, Nov.30-Dec.1, pp.39-46, ISBN 0-89791-759-6, ACM, 1995.

[Chisholm 1963] R. M. Chisholm, "Contrary-to-Duty Imperatives and Deontic Logic", *Analysis*, vol.24, pp.33-36, October, 1963.

[Cholvy & Cuppens 1997] L. Cholvy, F. Cuppens, "Analyzing Consistency of Security Policies", in *IEEE Symposium on Security and Privacy,* Oakland, California, May 4-7, pp.103-112, ISBN 0-8186-7828-3, IEEE Computer Society Press, 1997.

[Clark & Wilson 1987] D. Clark, D. Wilson, "A Comparison of Commercial and Military Computer Security Policies", in *IEEE Symposium on Security and Privacy,* Oakland, California, April 27-29, pp.184-194, ISBN 0-8186-0771-8, IEEE Computer Society Press, 1987.

[CLUSIF 1997] CLUSIF, *Evaluation des conséquences économiques des incidents et sinistres relatifs aux systèmes informatiques*, *France, 1996*, 13p., Club de la Sécurité Informatique Français, D-9701, février, 1997.

[CNIL 1978] CNIL, *"Loi n°78.17 du 6 Janvier 1978 relative à l'informatique, aux fichiers et aux libertés",* 1978.

[CNIL 1994] CNIL, *"Loi n°94-548 du 1° juillet 1994 relative au traitement de données nominatives ayant pour fin la recherche dans le domaine de la santé",* 1994.

[CO 1979] CO, *"Articles 47 et 48 du Code de Déontologie Médicale",* 1979.

[Corneillie *et al*. 1997] P. Corneillie, Y. Deswarte, A. Hawes, M. Kaâniche, H. Kurth, T. Manning, S. Moreau, A. Steinacker, *SQUALE - Definition of Draft Criteria for the Assessment of Dependable Systems - Draft 2*, ACTS Programme of the European Commission, ACTS95/AC097, LAAS Report 97166, May 15, 1997.

[CTCPEC 1993] CTCPEC, *The Canadian Trusted Computer Product Evaluation Criteria*, Canadian System Security Center, Communications Security Establishment, Governement of Canada, version 3.0e, January, 1993.

[Cuppens 1993a] F. Cuppens, "A Logical Analysis of Authorized and Prohibited Information Flows", in *IEEE Symposium on Research in Security and Privacy,* Oakland, California, May 24-26, pp.100-109, ISBN 0-8186-3370-0, IEEE Computer Society Press, 1993.

[Cuppens 1993b] F. Cuppens, "A Logical Formalization of Secrecy", in *Computer Security Foundations Workshop VI,* Franconia, June 15-17, pp.53-62, ISBN 0-8186-3950-4, IEEE Computer Society Press, 1993.

[Cuppens 1994a] F. Cuppens, "A Normative Framework for Security Policies", in *Towards a Global Expert System in Law (Conference in Celebration of the 25th Anniversary of the Istituto per la documentazione giuridica of the Consiglio Nationale delle Ricerche),* Firenze, Italy, 1994.

[Cuppens 1994b] F. Cuppens, "Roles and Deontic Logic", in *Second International Workshop on Deontic Logic in Computer Science,* Oslo, Norway, 1994.

[Cuppens & Saurel 1996] F. Cuppens, C. Saurel, "Specifying a Security Policy: A Case Study", in *9th IEEE Computer Security Foundations Workshop,* Kenmare, Ireland, June 10-12, pp.123-134, ISBN 0-8186-7522-5, IEEE Computer Society Press, 1996.

[Cuppens & Trouessin 1994] F. Cuppens, G. Trouessin, "Information Flow Controls vs Inference Controls: An Integrated Approach", in *Third European Symposium on Research in Computer Security (ESORICS 94),* (D. Gollman, Ed.), Brighton, United Kingdom, November, Lecture Notes in Computer Science, 875, pp.447-468, ISBN 3-540-58618-0, Springer-Verlag, 1994.

[d'Ausbourg 1994] B. d'Ausbourg, "Implementing Secure Dependencies over a Network by Designing a Distributed Security SubSystem", in *Third European Symposium on Research in Computer Security (ESORICS 94),* (D. Gollman, Ed.), Brighton, United Kingdom, Lecture Notes in Computer Science, pp.249-266, ISBN 3-540-58618-0, Springer-Verlag, 1994.

[Dacier 1993] M. Dacier, "A Petri Net Representation of the Take-Grant Model", in *Computer Security Foundations Workshop VI,* Franconia, USA, June 15-17, pp.99-108, ISBN 0-8186-3950-4, IEEE Computer Society Press, 1993.

[Dacier 1994] M. Dacier, *Vers une évaluation quantitative de la sécurité informatique,* Thèse de doctorat, Institut National Polytechnique de Toulouse, no.971, 145p., 20 décembre, 1994. (Rapport LAAS 94488.)

[Dacier & Deswarte 1994] M. Dacier, Y. Deswarte, "Privilege Graph: an Extension to the Typed Access Matrix Model", in *Third European Symposium on Research in Computer Security (ESORICS 94),* (D. Gollman, Ed.), Brighton, United Kingdom, Lecture Notes in Computer Science, 875, pp.317-334, ISBN 3-540-58618-0, Springer-Verlag, 1994.

[Dacier *et al*. 1996] M. Dacier, Y. Deswarte, M. Kaâniche, "Models and Tools for Quantitative Assessment of Operational Security", in *12th IFIP Information Systems Security Conference (IFIP/SEC'96),* (S. K. Katsikas, D. Gritzalis, Eds.), Samos, Greece, May 21-24, pp.177-186, ISBN 0-412-78120-4, Chapman & Hall, 1996.

[David 1995] J. David, "Organizational Security - Clean Up or Cover Up ?", *Computer & Security*, vol.14, no.2, pp.99-101, 1995.

[Davis 1993] A. M. Davis, *Software Requirements : Objects, Functions, and States,* 521p., ISBN 0-13-805763-X, Prentice Hall, 1993.

[DeMarco 1979] T. DeMarco, *Structured Analysis and System Specification,* Prentice Hall, 1979.

[Deswarte *et al*. 1991] Y. Deswarte, L. Blain, J.-C. Fabre, "Intrusion Tolerance in Distributed Computing Systems", in *IEEE Symposium on Research in Security and Privacy,* Oakland, California, May 20-22, pp.110-121, ISBN 0-8186-2168-0, IEEE Computer Society Press, 1991.

[DGS 1995] DGS, *"Circulaire DGS/DH n°95-22 du 6 mai 1995 relative aux droits des patients hospitalisés et comportant une charte du patient hospitalisé",* 1995.

[Dominus 1998] M.-J. Dominus, "Perl: Not Just for Web Programming", *IEEE Software*, pp.69-74, January-February, 1998.

[Emerson 1990] E. A. Emerson, "Temporal and Modal Logic", in *Handbook of Theoretical Computer Science, Formal Models and Semantics*, (J. van Leeuwen, Ed.), vol. B, pp.995-1072, ISBN 0444-88074-7, Elsevier, 1990.

[Fabre *et al*. 1996] J.-C. Fabre, Y. Deswarte, L. Blain, "Tolérance aux fautes et sécurité par fragmentation-redondance-dissémination. Fault tolerance and security by fragmentation-redundancy-scattering", *Technique et science informatiques*, vol.15, no.4, pp.405-427, 1996.

[Fariñas del Cerro & Herzig 1995] L. Fariñas del Cerro, A. Herzig, "Modal Deduction with Applications in Epistemic and Temporal Logic", in *Handbook of Logic in Artificial Intelligence and Logic Programming, Epistemic and Temporal Reasoning*, (D. M. Gabbay, C. J. Hogger, J. A. Robinson, Eds.), vol. 4/5, pp.499-594, ISBN 0-19-853791-3, Oxford Science Publications, 1995.

[Farmer & Spafford 1990] D. Farmer, E. H. Spafford, "The COPS Security Checker System", in *the Summer Usenix Conference,* Anaheim, CA, USA, 1990.

[Farmer & Spafford 1994] D. Farmer, E. H. Spafford, *The COPS Security Checker System*, Purdue University, Technical report, CSD-TR-94-993, January 22, 1994.

[Federal Criteria 1992] Federal Criteria, *Federal Criteria for Information Technology Security*, National Institute of Standards and Technology (NIST) and National Security Agency (NSA), Volume I and II, Draft, 1992.

[Fitting 1983] M. Fitting, *Proof Methods for Modal and Intuitionistic Logics,* Synthese Library (169), D. Reidel Publishing Company, Dordrecht, 1983.

[Fitting 1988] M. Fitting, "First-Order Modal Tableaux", *Journal of Automated Reasoning*, vol.4, no.2, pp.191-213, 1988.

[Fitting 1993] M. Fitting, "Basic Modal Logic", in *Handbook of Logic in Artificial Intelligence and Logic Programming, Logical Foundations*, (D. M. Gabbay, C. J. Hogger, J. A. Robinson, Eds.), vol. 1/5, pp.365-448, ISBN 0-19-853745-X, Oxford Science Publications, 1993.

[Fröhlich & Werner 1994] M. Fröhlich, M. Werner, "Demonstration of the Interactive Graph Visualization System daVinci", in *DIMACS Workshop on Graph Drawing,* (R. Tamassia, I. Tollis, Eds.), Princeton, USA, 1995, Lecture Notes in Computer Science (LNCS), 894, Springer-Verlag, 1994.

[Fröhlich & Werner 1996] M. Fröhlich, M. Werner, *"daVinci V2.0.x Online Documentation",* Universität Bremen, 1996.

[Garfinkel & Spafford 1996] S. Garfinkel, E. Spafford, *Practical Unix & Internet Security,* 2nd edition, 971p., ISBN 1-56592-148-8, O'Reilly & Associates (Inc.), 1996.

[Garvey & Lunt 1992] T. D. Garvey, T. F. Lunt, "Cover Stories for Database Security", in *Database Security, V: Status and Prospects,* (C. E. Landwehr, S. Jajodia, Eds.), Shepherstown, USA, November 4-7, 1991, pp.363-380, ISBN 0-444-89518-3, North-Holland, 1992.

[Glasgow *et al*. 1990] J. Glasgow, G. McEwen, P. Panangaden, "A Logic for Reasoning About Security", in *Computer Security Foundations Workshop,* Franconia, pp.2-13, IEEE Computer Society Press, 1990.

[Goguen & Meseguer 1982] J. Goguen, J. Meseguer, "Security Policies and Security Models", in *IEEE Symposium on Security and Privacy,* Oakland, California, pp.11-20, IEEE Computer Society, 1982.

[Haigh & Young 1986] J. T. Haigh, W. D. Young, "Extending the non-interference version of MLS for SAT", in *IEEE Symposium on Security and Privacy,* Oakland, California, April 7-9, pp.232-239, ISBN 0-8186-0716-5, IEEE Computer Society Press, 1986.

[Harrison & Ruzzo 1978] M. Harrison, W. Ruzzo, "On Synchronization and Security", in *Monotonic Protection Systems*, (R. DeMillo, D. Dobkin, A. Jones, R. Lipton, Eds.), pp.367-385, Academic Press, New York, 1978.

[Harrison *et al*. 1976] M. A. Harrison, W. L. Ruzzo, J. D. Ullman, "Protection in Operating Systems", *Communications of the ACM*, vol.19, no.8, pp.461-470, 1976.

[Heydon *et al*. 1990] A. Heydon, M. Maimone, J. Tygar, J. Wing, A. Zaremski, "Mirò: Visual Specification of Security", *IEEE Transactions on Software Engineering*, vol.16, no.10, pp.1185-1197, 1990.

[Hilpinen 1993] R. Hilpinen, "Actions in Deontic Logic", in *Deontic Logic in Computer Science*, (J.-J. C. Meyer, R. J. Wieringa, Eds.), pp.85-100, ISBN 0-471-93743-6, John Wiley & Sons, Chichester, England, 1993.

[Hughes & Cresswell 1968] G. E. Hughes, M. J. Cresswell, *An Introduction to Modal Logic,* University Paperbacks, 2nd edition, 388p., ISBN 0-415-04313-1, Routledge, 1968.

[ITSEC 1991] ITSEC, *Critères d'évaluation de la sécurité des systèmes informatiques,* v1.2, 163p., ISBN 92-826-3005-6, Office des publications officielles des Communautés Européennes, Luxembourg, 1991.

[ITSEM 1993] ITSEM, *Manuel d'évaluation de la sécurité des technologies de l'information,* v1.0, 262p., ISBN 92-826-7087-2, Commission des Communautés Européennes, Directorat Général XIII, Directorat B6, Luxembourg, 1993.

[JCSEC 1992] JCSEC, *The Japanese Computer Security Evaluation Criteria - Functionality Requirements*, Ministry of International Trade and Industry, Draft V1.0, August, 1992.

[Jones & Sergot 1992] A. J. I. Jones, M. Sergot, "Formal Specification of Security Requirements using the Theory of Normative Positions", in *Second European Symposium On Research In Computer Security (ESORICS 92),* (Y. Deswarte, G. Eizenberg, J.-J. Quisquater, Eds.), Toulouse, France, November 23-25, Lecture Notes in Computer Science, 648, pp.103-121, ISBN 3-540-56246-X & 0-387-56246-X, Springer-Verlag, 1992.

[Jones & Sergot 1993] A. J. I. Jones, M. Sergot, "On the Characterization of Law and Computer Systems: The Normative Systems Perspective", in *Deontic Logic in Computer Science*, (J.-J. C. Meyer, R. J. Wieringa, Eds.), pp.275-307, ISBN 0-471-93743-6, John Wiley & Sons, 1993.

[Jones *et al*. 1976] A. K. Jones, R. J. Lipton, L. Snyder, "A Linear Time Algorithm for deciding Security", in *17th Annual Symposium on Fundations of Computer Science,* Houston, USA, pp.33-41, 1976.

[Jonsher & Gerhardt 1991] D. Jonsher, W. Gerhardt, "A Role-Based Modelling of Access Control with the Help of Frames", in *7th International Conference and Exhibition on Information Security,* Brighton (UK), May, pp.131-142, 1991.

[Joubert *et al*. 1982] M. Joubert, M. Fieschi, D. Fieschi, M. Roux, "Medical Decision Aid: Logic Bases of the System SPHINX", in *First International Logic Programming Conference,* Marseille, France, September, 14-17, pp.210-214, 1982.

[Kanger 1972] S. Kanger, "Law and Logic", *Theoria*, vol.38, no.3, pp.105-132, 1972.

[Karger & Wray 1991] P. A. Karger, J. C. Wray, "Storage Channels in Disk Arm Optimization", in *IEEE Symposium on Research in Security and Privacy,* Oakland, California, May 20-22, pp.52-61, ISBN 0-8186-2168-0, IEEE Computer Society Press, 1991.

[Kohl & Neuman 1993] J. Kohl, C. Neuman, *The Kerberos Network Authentification Service (V5)*, Internet RFC 1510, 1993.

[Kripke 1959] S. Kripke, "A Completeness Theorem in Modal Logic", *Journal of Philosophical Logic*, vol.24, pp.1-14, 1959.

[Kripke 1963] S. A. Kripke, "Semantical Considerations in Modal Logic", *Acta Philosophica Fennica*, vol.16, pp.83-94, 1963.

[Kuper 1987] G. Kuper, "Logic Programming with Sets", in *6th ACM Conference on Principles of Database Systems (PODS),* San Diego, California, USA, March 23-25, pp.11-20, ISBN 0-89791-223-3, ACM Press, 1987.

[Kuper 1988] G. Kuper, "On the Expressive Power of Logic Programming with Sets", in *7th ACM Conference on Principles of Database Systems (PODS),* Austin, Texas, USA, March 21-23, pp.10-14, ISBN 0-89791-263-2, ACM Press, 1988.

[Laffont & Ortalo 1997] J. Laffont, R. Ortalo, *Editeur de politiques de sécurité utilisant le formalisme des logiques modales*, LAAS-CNRS, Rapport 97064, 1997.

[Lampson 1971] B. Lampson, "Protection", *5th Princeton Symposium on Information Sciences and Systems*, 1971.

[Laprie 1995] J.-C. Laprie (Ed.), *Guide de la Sûreté de Fonctionnement,* 2° edition, 324 p., ISBN 2-85428-341-4, Cépaduès Editions, Toulouse, France, 1995.

[Lawrence 1993] L. G. Lawrence, "The Role of Roles", *Computers & Security*, vol.12, pp.15-21, 1993.

[Lee 1989] T. Lee, "Statistical Models of Trust: TCB's vs. People", in *IEEE Symposium on Security and Privacy,* Oakland, California, May 1-3, pp.10-19, ISBN 0-8186-1939-2, IEEE Computer Society Press, 1989.

[Lindahl 1977] L. Lindahl, *Position and Change - A Study in Law and Logic,* Synthese Library (112), D. Reidel, Dordrecht, 1977.

[Lipton & Snyder 1978] R. Lipton, L. Snyder, "On Synchronization and Security", in *Foundations of Secure Computation*, (R. DeMillo, D. Dobkin, A. Jones, R. Lipton, Eds.), pp.367-385, Academic Press, New York, 1978.

[Littlewood *et al*. 1993] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid, D. Gollmann, "Towards Operational Measures of Computer Security", *Journal of Computer Security*, vol.2, no.2-3, pp.211-229, 1993.

[Lotz 1997] V. Lotz, "Threat Scenarios as a Means to Formally Develop Secure Systems", *Journal of Computer Security*, vol.5, no.1, pp.31-67, 1997.

[Maibaum 1993] T. Maibaum, "Temporal Reasoning over Deontic Specifications", in *Deontic Logic in Computer Science*, (J.-J. C. Meyer, R. J. Wieringa, Eds.), pp.141-202, ISBN 0-471-93743-6, Wiley, Chichester, England, 1993.

[McCullough 1987] D. McCullough, "Specifications for Multi-Level Security and a Hook-Up Property", in *IEEE Symposium on Security and Privacy,* Oakland, California, April 27-29, pp.161-166, ISBN 0-8186-0771-8, IEEE Computer Society Press, 1987.

[McCullough 1990] D. McCullough, "A Hookup Theorem for Multilevel Security", *IEEE Transactions on Software Engineering*, vol.16, no.6, pp.563-568, June, 1990.

[McLean 1987] J. McLean, "Reasoning about Security Models", in *IEEE Symposium in Security and Privacy,* Oakland, California, April 27-29, pp. 123-131, ISBN 0-8186-0771-8, IEEE Computer Society Press, 1987.

[McLean 1990] J. McLean, "Security Models and Information Flow", in *IEEE Symposium on Research in Security and Privacy,* Oakland, California, pp. 180-187, ISBN 0-8186-2060-9, IEEE Computer Society Press, 1990.

[McLean 1994] J. McLean, "Security Models", in *Encyclopedia of Software Engineering*, (J. J. Marciniak, Ed.), vol. 2/2, pp.1136-1145, John Wiley & Sons, Inc., 1994.

[McLean *et al*. 1984] J. McLean, C. E. Landwehr, C. L. Heitmeyer, "A Formal Statement of the MMS Security Model", in *IEEE Symposium on Security and Privacy,* Oakland, California, April 29-May 2, pp.188-194, ISBN 0-8186-0532-4, IEEE Computer Society Press, 1984.

[Meyer & Wieringa 1993] J.-J. C. Meyer, R. J. Wieringa (Eds.), *Deontic Logic in Computer Science,* 317 p., ISBN 0-471-93743-6, Jon Wiley & Sons, 1993.

[Millen 1987] J. K. Millen, "Covert Channel Capacity", in *IEEE Symposium on Security and Privacy,* Oakland, California, April 27-29, pp.60-66, ISBN 0-8186-0771-8, IEEE Computer Society Press, 1987.

[Minsky & Lockman 1985] N. H. Minsky, A. Lockman, "Ensuring Integrity by Adding Obligations to Privileges", in *8th International Conference on Software Engineering,* pp.92-102, 1985.

[Mohammed & Dilts 1994] I. Mohammed, D. M. Dilts, "Design for Dynamic User-Role-Based Security", *Computers & Security*, vol.13, no.8, pp.661-671, 1994.

[Morgenstern 1988] M. Morgenstern, "Controlling Logical Inference in Multilevel Database Systems", in *IEEE Symposium on Security and Privacy,* Oakland, California, April 18-21, pp.245-255, ISBN 0-8186-0850-1, IEEE Computer Society Press, 1988.

[Moskowitz 1992] I. S. Moskowitz, "The Influence of Delay upon an Idealized Channel's Bandwidth", in *IEEE Symposium on Research in Security and Privacy,* Oakland, California, May 4-6, pp.62-67, ISBN 0-8186-2825-1, IEEE Computer Society Press, 1992.

[Moskowitz & Kang 1997] I. S. Moskowitz, M. H. Kang, "An Insecurity Flow Model", in *New Security Paradigms 1997 Workshop,* September, 1997.

[Moskowitz & Miller 1994] I. S. Moskowitz, A. R. Miller, "Simple Timing Channels", in *IEEE Symposium on Research in Security and Privacy,* Oakland, California, May 16-18, pp.56-64, ISBN 0-8186-5675-1, IEEE Computer Society Press, 1994.

[Muffet 1992] A. D. E. Muffet, *"Crack Version 4.1 — A Sensible Password Checker for Unix",* publicly available by ftp with the `crack4.1` software at `ftp.cert.org`, 1992.

[Myers *et al*. 1997a] B. A. Myers, E. Borison, A. Ferrency, R. McDaniel, R. C. Miller, A. Faulring, B. D. Kyle, P. Doane, A. Mickish, A. Klimovitski, *The Amulet V3.0 Reference Manual*, 396p., Carnegie Mellon University, CMU-CS-95-166-R2 & CMU-HCII-95-102-R2, March, 1997.

[Myers *et al*. 1997b] B. A. Myers, R. G. McDaniel, R. C. Miller, A. S. Ferrency, A. Faulring, B. D. Kyle, A. Mickish, A. Klimovitski, P. Doane, "The Amulet Environment: New Models for Effective User Interface Software Development", *IEEE Transations on Software Engineering*, vol.23, no.6, pp.347-365, June, 1997.

[Nicomette 1996] V. Nicomette, *La protection dans les systèmes à objets répartis,* Thèse de doctorat, Institut National Polytechnique de Toulouse, no. 1252, 177 p., 17 décembre, 1996. (Rapport LAAS 96496.)

[Nicomette & Deswarte 1997] V. Nicomette, Y. Deswarte, "An Authorization Scheme For Distributed Object Systems", in *IEEE Symposium on Security and Privacy,* Oakland, California, May 4-7, pp.21-30, ISBN 0-8186-7828-3, IEEE Computer Society Press, 1997.

[Olovsson *et al*. 1995] T. Olovsson, E. Jonsson, S. Brocklehurst, B. Littlewood, "Towards Operational Measures of Computer Security: Experimentation and Modelling", in *Predictably Dependable Computing Systems*, (B. Randell, J.-C. Laprie, H. Kopetz, B. Littlewood, Eds.), ESPRIT Basic Research Series, pp. 555-569, ISBN 3-540-59334-9, Springer-Verlag, Berlin, Germany, 1995.

[Orr 1981] K. Orr, *Structured Requirements Definition,* Ken Orr and Associates, Topeka, Kansas, 1981.

[Ortalo 1997] R. Ortalo, *Using Role-Based Abstractions for Security Policy Specification with Deontic Logic*, 20p., LAAS-CNRS, Rapport 97216, June 1997.

[Ortalo 1998] R. Ortalo, "A Flexible Method for Information System Security Policy Specification", in *5th European Symposium on Research in Computer Security (ESORICS 98),* Louvain-la-Neuve, Belgique, September 16-18, Lecture Notes in Computer Science, Springer-Verlag, 1998. (À paraître. — Rapport LAAS 98079.)

[Ortalo & Deswarte 1998a] R. Ortalo, Y. Deswarte, "Management of Information System Security: Specification and Assessment", in *14th International Conference on Advanced Science and Technology,* (L. Henschen, Ed.), Naperville, Illinois, USA, April 3-4, pp.207-221, Library of Congress 98-84876, CAPAMA, 1998. (Rapport LAAS 97567.)

[Ortalo & Deswarte 1998b] R. Ortalo, Y. Deswarte, "Quantitative Evaluation of Information System Security", in *14th IFIP International Information Security Conference (IFIP/SEC'98),* Vienna-Budapest, Austria-Hungary, August 31-September 4, Chapman & Hall, 1998. (À paraître. — Rapport LAAS 98107.)

[Ortalo *et al*. 1997] R. Ortalo, Y. Deswarte, M. Kaâniche, "Experimenting with Quantitative Evaluation Tools for Monitoring Operational Security", in *Dependable Computing for Critical Applications 6 (DCCA'6),* (M. Dal Cin, C. Meadows, W. H. Sanders, Eds.), Grainau, Germany, March 5-7, Dependable Computing and Fault-Tolerant Systems, vol.11, pp.307-328, ISBN 0-8186-8009-1, IEEE Computer Society Press, 1997. (Rapport LAAS 96369.)

[Pigeaud 1993] G. Pigeaud, *"Consentement : l'art difficile de la communication",* Objectif Soins, no.17, pp.20-21, Novembre 1993.

[Pörn 1977] I. Pörn, *Action Theory and Social Science: Some Formal Models,* Synthese Library (120), D. Reidel, Dordrecht, 1977.

[Reiter & Stubblebine 1997] M. K. Reiter, S. G. Stubblebine, "Towards Acceptable Metrics of Authentification", in *IEEE Symposium on Security and Privacy,* Oakland, California, May 4-7, pp.10-20, ISBN 0-8186-7828-3, IEEE Computer Society Press, 1997.

[Ross 1977] D. T. Ross, "Structured Analysis (SA): A Language for Communicating Ideas", *IEEE Transactions on Software Engineering*, vol.3, no.1, pp.16-34, January, 1977.

[Royakkers 1994] L. M. M. Royakkers, "Towards a Deontic Logic Approach to Legal Rules", in *Second International Workshop on Deontic Logic in Computer Science,* Amsterdam, pp.319-332, The Netherlands, 1994.

[Rushby 1993] J. Rushby, *Formal Methods and the Certification of Critical Systems*, 313p., SRI International, Menlo Park CA 94025, USA, Technical Report, CSL-93-7, December, 1993.

[Sandhu 1988] R. S. Sandhu, "The Schematic Protection Model: Its Definition and Analysis for Acyclic Attenuation Schemes", *Journal of the ACM*, vol.35, no.2, pp.404-432, 1988.

[Sandhu 1992] R. S. Sandhu, "Expressive Power of the Schematic Protection Model", *Journal of Computer Security*, vol.1, no.1, pp.59-98, 1992.

[Sandhu 1993] R. S. Sandhu, "Lattice-Based Access Control Models", *IEEE Computer*, vol.26, no.11, pp.9-19, November, 1993.

[Sandhu 1995] R. S. Sandhu, "Roles Versus Groups", in *1st ACM Workshop on Role-Based Access Control,* NIST, Gaithersburg, Maryland, USA, Nov.30-Dec.1, pp.25-26, ISBN 0-89791-759-6, ACM, 1995.

[Sandhu 1996] R. S. Sandhu, "Role Hierarchies and Constraints for Lattice-Based Access Controls", in *4th European Symposium on Research in Computer Security (ESORICS'96),* (E. Bertino, H. Kurth, G. Martella, E. Montolivo, Eds.), Rome, Italy, September 25-27, Lecture Notes in Computer Science, 1146, pp.65-79, ISBN 3-540-61770-1, Springer-Verlag, 1996.

[Sandhu *et al*. 1996] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman, "Role-Based Access Control Models", *IEEE Computer*, vol.29, no.2, pp.38-47, February, 1996.

[Santos & Carmo 1993] F. Santos, J. Carmo, "A Deontic Logic Representation of Contractual Obligations", in *Deontic Logic in Computer Science* (J.-J. C. Meyer, R. J. Wieringa, Eds.), pp.244-257, ISBN 0-471-93743-6, John Wiley & Sons, 1993.

[Saury 1991] R. Saury, *"Le Secret Médical",* Gestions hospitalières, no.303, pp.120-125, février 1991.

[Shmueli *et al*. 1988] O. Shmueli, S. Tsur, C. Zaniolo, "Rewriting of Rules Containing Set Terms in a Logic Data Language (LDL)", in *7th annual ACM symposium on Principles of Database Systems (PODS),* Austin, Texas, USA, pp.15-28, ISBN 0-89791-263-2, ACM, 1988.

[Snyder 1981] L. Snyder, "Theft and Conspiracy in the Take-Grant Model", *Journal of Computer and System Sciences*, vol.23, pp.333-347, 1981.

[Solms & Merwe 1994] S. H. v. Solms, I. v. d. Merwe, "The Management of Computer Security Profiles using a Role-oriented Approach", *Computers & Security*, vol.13, no.8, pp.673-680, 1994.

[Sutherland 1986] D. Sutherland, "A Model of Information", in *9th NIST/NCSC National Computer Security Conference,* 1986.

[TCSEC 1985] TCSEC, *Trusted Computer System Evaluation Criteria*, 122 p., Department of Defense (DoD), DoD Standard, DoD 5200.28-STD, 1985.

[TNI 1987] TNI, *Trusted Network Interpretation of the Trusted Computer Security Evaluation Criteria*, 278 p., National Computer Security Center, NCSC-TG-005, 1987.

[Trouessin 1991a] G. Trouessin, "Quantitative Evaluation of Confidentiality by Entropy Calculation", in *The Computer Security Foundations Workshop IV,* Franconia, USA, June 18-20, pp.12-21, ISBN 0-8186-2215-6, IEEE Computer Society Press, 1991.

[Trouessin 1991b] G. Trouessin, *Traitements Fiables de Données Confidentielles par Fragmentation-Redondance-Dissémination,* Thèse de Doctorat, Université Paul Sabatier, no.1077, 162 p., 20 décembre, 1991. (Rapport LAAS 91412.)

[Van Benthem 1983] J. Van Benthem, *The Logic of Time,* Synthese Library, 156, D. Reidel Publishing Company, 1983.

[Wall *et al*. 1996] L. Wall, T. Christiansen, R. L. Schwartz, *Programming Perl,* 2nd edition, 645 p., ISBN 1-56592-149-6, O'Reilly & Associates, 1996.

[Weissman 1992] C. Weissman, "BLACKER: Security for the DDN, Examples of A1 Security Engineering Trades", in *IEEE Symposium on Research in Security and Privacy,* Oakland, California, May 4-6, pp.286-292, ISBN 0-8186-2825-1, IEEE Computer Society Press, 1992.

[Yu & Gligor 1988] C.-F. Yu, V. E. Gligor, "A Formal Specification and Verification Method for the Prevention of Denial of Service", in *IEEE Symposium on Security and Privacy,* Oakland, California, April 18-21, pp.187-202, ISBN 0-8186-0850-1, IEEE Computer Society Press, 1988.

[Zakinthinos & Lee 1994] A. Zakinthinos, E. S. Lee, "The Composability of Non-Interference", *Journal of Computer Security*, vol.3, no.4, pp.269-281, 1994.