SYNTHÈSE

Détection d'intrusions : corrélation d'alertes

Hervé Debar[1] — Benjamin Morin[1] — Frédéric Cuppens[2,4]
Fabien Autrel[2] — Ludovic Mé[3] — Bernard Vivinis[3]
Salem Benferhat[4] — Mireille Ducassé[5] — Rodolphe Ortalo[6]

[1] *France Télécom R&D, Caen, France*
{herve.debar,benjamin.morin}@rd.francetelecom.com
[2] *ONERA, Toulouse, France*
frederic.cuppens@enst-bretagne.fr, autrel@cert.fr
[3] *Supélec, Rennes, France*
{Ludovic.Me,Bernard.Vivinis}@supelec.fr
[4] *IRIT, Toulouse, France*
frederic.cuppens@enst-bretagne.fr, benferhat@cril.univ-artois.fr
[5] *IRISA/INSA, Rennes, France*
ducasse@irisa.fr
[6] *Calyx NetSecure, France*
rodolphe.ortalo@cram-mp.fr

RÉSUMÉ. *Les outils de détection d'intrusions (IDS) actuellement en opération produisent de trop nombreuses alertes. Les informations qu'elles contiennent manquent de précision et sont, de plus, parcellaires et de très bas niveau. Ces alertes sont par conséquent d'un intérêt limité pour un opérateur humain. La recherche sur la corrélation d'alertes est très prometteuse. Par la corrélation d'alertes nous pouvons espérer réduire le volume d'informations à traiter, améliorer la qualité du diagnostic proposé et dégager une meilleure vision globale de l'état de sécurité du système en cas d'intrusion. Cet article de synthèse présente différentes techniques de corrélation d'alertes et décrit comment ces techniques peuvent être utilisées pour la détection d'intrusions.*

ABSTRACT. *Current intrusion detection systems generate too many alerts. These alerts are imprecise and partial. Furthermore, they contain low level information. These alerts are therefore of limited interest for a human operator. Alert correlation is a promising technology to reduce the number of alerts, improve the diagnostic and provide a better vision of the security of the system in the case of an intrusion. This paper presents an overview of different alert correlation technologies and shows how these technologies can be applied to intrusion detection.*

MOTS-CLÉS : *sécurité, détection d'intrusions, corrélation.*

KEYWORDS: *security, intrusion detection, correlation.*

1. Introduction

Les outils de détection d'intrusions actuellement en opération produisent de trop nombreuses alertes. Les informations qu'elles contiennent manquent de précision et sont, de plus, parcellaires et de très bas niveau. Ces alertes sont par conséquent d'un intérêt limité pour un opérateur humain. La corrélation d'alertes semble être une des clés de l'évolution des systèmes de détection d'intrusions. En effet, en corrélant les informations contenues dans les alertes, ainsi que d'éventuelles informations additionnelles, on peut espérer réduire le volume d'informations à traiter, améliorer la qualité du diagnostic proposé et dégager une meilleure vision globale de l'état de sécurité du système en cas d'intrusion.

Cet article de synthèse propose une introduction aux techniques de corrélation d'alertes appliquées dans le domaine de la détection d'intrusions. Il commence par introduire les concepts essentiels de la détection d'intrusions (section 2). Un panorama des objectifs possibles de la corrélation d'alertes permet ensuite de définir un ensemble de fonctions de corrélation intéressantes (section 3). Les fonctions et limites des produits existants sont ensuite présentées (section 4). Enfin, sont détaillées trois grandes familles de corrélation possédant des caractéristiques spécifiques : la première famille s'intéresse à la corrélation utilisant des données topologiques issues du système d'information surveillé (section 5) ; la deuxième famille s'intéresse à la corrélation utilisant les propriétés intrinsèques du flux d'alertes, dite corrélation implicite (section 6) ; finalement, la troisième famille décrit la corrélation utilisant la connaissance des scénarios d'attaque en association avec les alertes, dite corrélation explicite (section 7).

2. Les concepts essentiels de la détection d'intrusions

Cette section introduit des notions utilisées dans la suite de l'article. Plusieurs schémas ont été proposés pour décrire les composants d'un système de détection d'intrusions. Parmi eux, nous avons retenu celui issu des travaux du *Intrusion Detection exchange format Working Group* (IDWG) de l'Internet Engineering Task Force (IETF) comme base de départ, car il résulte d'un large consensus parmi les intervenants du domaine. Ces travaux sont présentés section 2.1. Les notions définies ne couvrent cependant pas complètement les besoins de la corrélation d'alertes. Nous raffinons l'architecture d'un système de détection d'intrusions section 2.2. Enfin nous introduisons les notions de corrélation d'alertes implicite et explicite section 2.3.

2.1. *Les travaux de l'IDWG*

L'objectif des travaux du groupe IDWG est la définition d'un standard de communication entre certains composants d'un système de détection d'intrusions. Comme illustré par la figure 1, l'architecture IDWG d'un système de détection d'intrusions contient des *capteurs* qui envoient des *événements* à un analyseur. Un ou des capteurs

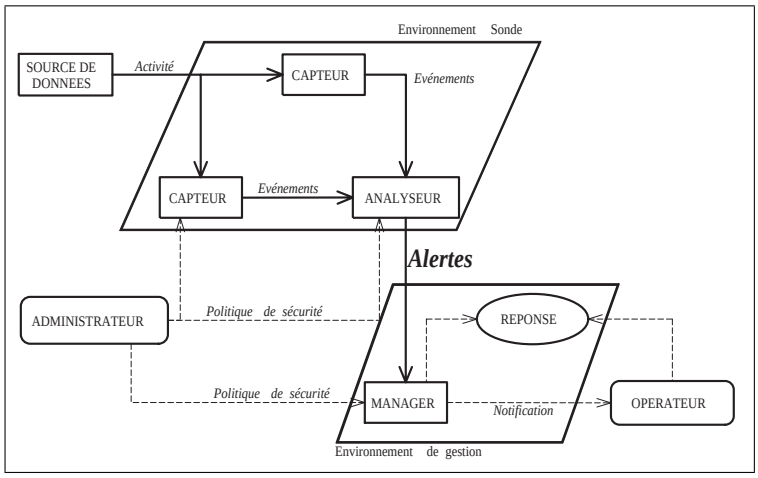

Figure 1. *Architecture IDWG d'un système de détection d'intrusions*

couplés avec un analyseur forment une *sonde*. Une sonde envoie des *alertes* vers un *manager* qui la notifie à un opérateur humain.

Il faut noter que le flux de communication effectivement standardisé par le document IDWG est le flux des alertes. Ce standard inclut la définition d'un format de message d'alertes et d'un protocole de communication sécurisé pour le transport de ces messages. Ce format et ce protocole sont très importants pour permettre à des outils hétérogènes de fonctionner dans un même environnement. Le détail de leur description n'est cependant pas pertinent pour la suite de cet article. Il peut être trouvé dans le document de définitions [WOO 02].

2.2. *Une architecture mieux adaptée à la corrélation d'alertes*

Un système de détection d'intrusions commercial moderne se décompose en plusieurs composants logiques échangeant au travers de flux d'informations. Un exemple d'architecture de cet ensemble de composants et de flux est représenté par la figure 2. Les types des composants sont au nombre de quatre : les sondes, les consoles de gestion, les concentrateurs d'alertes et les consoles d'alertes.

Les consoles de gestion : les consoles de gestion sont utilisées pour gérer les sondes, particulièrement la mise à jour des configurations, la mise à jour des signatures et la mise à jour des logiciels en cas de nécessité. Le premier flux d'informations est le flux de gestion interne du système de détection d'intrusions ; il transporte les informations précitées.

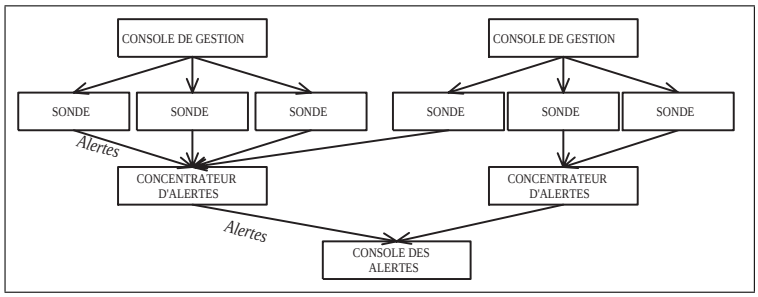

Figure 2. *Une architecture mieux adaptée à la corrélation d'alertes*

Les sondes : les sondes sont chargées d'analyser le flux de données et d'émettre des alertes à destination des concentrateurs.

Les concentrateurs d'alertes : les concentrateurs d'alertes prennent en charge des fonctions de corrélation d'alertes entre les outils de détection d'intrusions et d'autres outils comme les analyseurs de vulnérabilités et les gardes-barrière (non indiqués sur le schéma). De plus, ils offrent un espace de stockage ce qui permet de répartir la charge réseau et donc de déployer un nombre de sondes plus important.

Les consoles d'alertes : les consoles d'alertes permettent la visualisation et le traitement des informations fournies par le système de détection d'intrusions. Ces consoles fonctionnent souvent en mode interactif, les alertes étant affichées en temps réel aux opérateurs. Cet affichage en temps réel est souvent la partie la plus visible des diagnostics proposés, mais ce n'est pas nécessairement la plus utile. En effet, la console doit permettre de naviguer entre les différentes alertes. Pour chacune de ces alertes un contexte doit être rappelé, par exemple il est important de connaître les autres alertes provenant de la même source (une source est identifiée le plus souvent par son adresse IP). La vue d'ensemble ainsi fournie permet d'affiner le diagnostic. Une autre fonction utile est l'évaluation de l'alerte elle-même. En effet, les fausses alarmes liées au fonctionnement du système d'information surveillé se produisent de manière récurrente et peuvent être facilement et rapidement traitées par l'opérateur.

On peut remarquer que la corrélation d'alertes peut être effectuée sur les sondes comme sur les concentrateurs d'alertes. La différence essentielle entre ces deux localisations est le fait que la corrélation d'alertes sur les sondes n'utilise qu'une seule source de données, alors que la corrélation d'alertes sur les concentrateurs s'applique à de multiples sources de données possiblement hétérogènes.

2.3. *Approches de la corrélation d'alertes*

Les travaux autour de la corrélation d'alertes dans le domaine de la détection d'intrusions sont relativement récents. Ces travaux sont issus d'observations et d'expérimentations terrain ; la base théorique est encore en construction aujourd'hui. Dans la littérature, on peut toutefois identifier deux approches principales pour traiter la corrélation.

Corrélation implicite : l'exploitation des alertes révèle des relations intrinsèques entre elles. Une relation peut être constituée par exemple par une correspondance fréquentielle ou statistique entre des alertes. Cette correspondance est obtenue par une analyse automatique des données.

Corrélation explicite : l'opérateur est capable d'exprimer explicitement des relations entre différentes alertes, sous la forme d'un scénario. Un scénario regroupe en général un ensemble de propriétés que doivent satisfaire les alertes, et des liens les connectant.

Ces deux formes de corrélation sont à rapprocher des deux approches classiques de la détection d'intrusions : l'approche comportementale et l'approche par scénario.

Dans le premier cas, une alerte est générée dès qu'est détecté un comportement non conforme à une référence, généralement construite par apprentissage. Or, la déviation du comportement observée peut être due à une évolution naturelle de l'environnement et du système : c'est un faux positif (fausse alarme). En outre, l'attaquant (utilisateur interne malicieux) peut modifier lentement son comportement afin de parvenir à un comportement intrusif qui, ayant été progressivement appris, ne sera pas détecté : c'est un faux négatif (absence d'alarme en présence d'attaque).

Le risque de faux positifs est moindre avec l'approche par scénario car toute activité considérée comme litigieuse est décrite dans une base de signatures d'attaque. Une alarme est émise dès qu'une activité observée est conforme à une des signatures. La qualité de la signature est importante : si elle n'est pas assez précise, elle peut également conduire à de nombreux faux positifs. En outre, bien évidemment, si la signature de l'attaque n'est pas dans la base (comme c'est le cas pour les nouvelles attaques), l'attaque en question ne sera pas détectée (c'est là un problème similaire à ce que l'on connaît avec les bases de signatures de virus). La maintenance de la base est donc essentielle.

Les corrélations implicite et explicite présentent les mêmes types d'inconvénient et d'avantage.

Nous présentons des techniques de corrélation implicite section 6 et des techniques de corrélation explicite section 7.

Les deux formes de corrélation exploitent des données contenues dans le flux d'alertes et, potentiellement, dans une base d'informations additionnelle, par exemple décrivant la topologie. La section 5 détaille ce qui peut être obtenu en exploitant ce type de sources de données additionnelles.

3. Objectifs de la corrélation d'alertes

Cette section identifie des objectifs possibles de la corrélation d'alertes. Il faut noter qu'aucun des outils existants, commerciaux ou prototypes de recherche, ne couvre tous ces objectifs. Nous retenons ici trois objectifs principaux de la corrélation d'alertes : la réduction du volume d'informations à traiter par les opérateurs et les analystes, l'augmentation de la qualité du diagnostic fourni, le suivi des attaques au cours du temps. Chacun de ces trois objectifs est détaillé dans les paragraphes 3.1, 3.2 et 3.3.

3.1. *Réduction du volume d'informations*

Le premier objectif de la corrélation d'alertes en détection d'intrusions est de réduire la quantité d'alertes que l'opérateur doit traiter. En effet, une sonde de détection d'intrusions peut générer un grand nombre d'alertes par unité de temps, alors que la capacité de traitement d'un opérateur est limitée.

Cette réduction est d'autant plus importante que l'expérience montre que dans chaque environnement, certaines alertes surviennent fréquemment, pour des raisons liées à la configuration et au fonctionnement du système d'information surveillé. Ces alertes peuvent être désactivées dans la configuration du système de détection d'intrusions, mais cela n'est pas désirable si des attaques peuvent en conséquence ne plus être détectées. Un outil de corrélation peut donc automatiser le contrôle des alertes reçues pour vérifier si elles sont intéressantes.

Plus précisément, l'objectif de réduction du volume d'information se décline suivant quatre sous-objectifs :

L'élimination : pour un même événement dans la source d'informations, plusieurs alertes redondantes peuvent être générées par une ou plusieurs sondes de détection d'intrusions. Par exemple, une requête HTTP unique peut donner lieu à plusieurs alertes si plusieurs sondes détectent la même tentative malveillante simultanément. Dans ce cas, on peut ne garder qu'une seule de ces alertes.

La fusion : plusieurs alertes peuvent être fusionnées en une seule qui contient l'ensemble des informations portées par ces alertes. Par exemple, certaines attaques se caractérisent par des rafales d'événements. Certains systèmes de détection d'intrusions génèrent une alerte par événement dans la rafale. Il est donc souhaitable de

fusionner les alertes liées à cette rafale (par exemple par comptage) pour faciliter le traitement de l'information.

L'agrégation : face à un grand nombre d'alertes, il est parfois possible de dégager des caractéristiques majoritaires communes à un sous-ensemble de ces alertes. Ces caractéristiques majoritaires peuvent par exemple être l'adresse de l'attaquant, l'adresse de la cible ou le type d'attaque pratiqué. L'information présentée par l'ensemble agrégé permet de faire un traitement d'ensemble sans s'intéresser individuellement à chaque alerte.

La synthèse : plusieurs alertes peuvent être liées par exemple par des règles logiques explicites (par exemple des scénarios d'attaque connus) ou par des lois statistiques apprises. Il est possible en utilisant cette connaissance de synthétiser un ensemble d'alertes en une alerte résumant cette connaissance.

Suivant les cas, les alertes surnuméraires sont simplement éliminées du flux d'alertes, remplacées par l'alerte équivalente ou conservées à côté de l'alerte équivalente.

3.2. *Amélioration de la qualité du diagnostic*

Une alerte fournie par un système de détection d'intrusions est aujourd'hui souvent de très bas niveau et contient peu d'informations au sujet de la vulnérabilité exploitée ou de l'anomalie qu'elle décrit.

En particulier, les informations suivantes sont intéressantes pour évaluer la sévérité de l'attaque et sont, actuellement, souvent omises.

Vulnérabilité effective : l'action de l'attaquant correspond à une vulnérabilité effectivement présente dans le système d'information.

Vulnérabilité passée : l'action de l'attaquant correspond à une vulnérabilité présente dans le système d'information dans le passé ; cette vulnérabilité a été supprimée, par exemple suite à une mise-à-jour, et le système d'information n'est plus vulnérable au moment où l'attaque est exécutée.

Vulnérabilité impossible : le système attaqué n'a aucune relation avec la tentative de l'attaquant. Par exemple, la vulnérabilité concerne un serveur IIS et le système attaqué est un serveur Apache.

Prise d'empreinte : l'action de l'attaquant lui permet de déterminer si le système d'information est vulnérable ou non.

Attaque caractérisée : l'action de l'attaquant est capable, en cas de succès, de compromettre le système d'information.

Réussite : l'attaquant a obtenu ce qu'il cherchait par son action, soit l'information recherchée, soit la compromission du système visé.

D'autres critères intéressants peuvent concerner l'identification d'outils d'attaque ou la propagation des attaquants.

L'augmentation de la qualité du diagnostic peut, en particulier, être réalisée par les techniques d'agrégation et de synthèse mentionnées ci-dessus.

3.3. *Suivi des attaques*

Une attaque se caractérise rarement par une action isolée et elle est généralement constituée par plusieurs sous-attaques. Un attaquant cherche en général à obtenir des renseignements sur le système d'information cible, ses différents composants physiques et logiques, les services qu'il publie officiellement à l'extérieur et ceux qui peuvent exister par ailleurs. En fonction de ces informations, l'attaquant va choisir un certain nombre de cibles, les attaquer et obtenir des résultats qui guideront son évolution. De même chaque sous-attaque peut donner lieu, vu de l'IDS, à plusieurs alertes.

Dans cette optique, l'objectif de la corrélation d'alertes est d'abord d'identifier les sous-attaques par corrélation des alertes qu'elles génèrent. Puis, il faut être capable d'identifier les sous-attaques constitutives de l'attaque complète.

Cette identification doit être à la fois rapide (pour permettre d'agréger un petit nombre d'alertes) et robuste (pour résister aux mutations de l'apparence de cette identité réalisées intentionnellement par l'attaquant ou imposées par son environnement). Pour chaque attaquant identifié, la corrélation d'alertes doit permettre d'archiver et de classer les actions réalisées, les informations accumulées sur l'attaquant et les informations que les cibles ont fournies.

4. Fonctions et limites de produits existants

Lors du déploiement des premiers outils de détection d'intrusions, il est apparu assez vite que la quantité d'alertes générées était importante et demandait un environnement séparé pour la gestion de ces alertes. De ce besoin sont nées les premières plate-formes de gestion d'alertes ainsi que l'architecture présentée dans la figure 2. La gestion d'alertes effectuées par les outils existants recouvre essentiellement trois activités. Tout d'abord, les outils commencent par centraliser des traces (section 4.1). Certains outils croisent les alertes et les rapports d'audit fournis par les analyseurs de vulnérabilité (section 4.2). Enfin, un outil agrège les alertes en temps réel (section 4.3).

Les informations contenues dans cette section s'appuient sur une évaluation des outils commerciaux dont le détail peut être trouvé dans [DEB 02].

4.1. *Centralisation de traces*

Les outils commencent d'abord par centraliser les alertes générées par les sondes. Cette centralisation répond d'abord à un besoin de déploiement. En effet, les sondes de détection d'intrusions génèrent un grand nombre d'alertes qu'il n'est pas souhaitable de transporter en temps réel, pour éviter les coûts d'établissement de communications et la consommation de ressources réseau parfois limitées. En utilisant des concentrateurs d'alertes comme stockage et relais, cela permet de faciliter le déploiement d'un grand nombre de sondes.

Une fois cette base de stockage intermédiaire mise en œuvre, les vendeurs ont réalisé qu'elle pouvait être utilisée pour des besoins de stockage de traces en général. Très rapidement, ces concentrateurs ont été utilisés pour collecter des sources diverses, par exemple des messages syslog, des traces de connexion de serveurs DHCP ou des traces de *firewall*. Cette approche vise à créer une base de données permettant de recevoir des traces que l'administrateur du système d'information souhaite conserver, consulter et croiser facilement. Il s'agit donc d'une approche permettant la juxtaposition d'alertes hétérogènes, mais n'incluant pas encore de fonction de corrélation.

La majorité des produits commerciaux, tels NetSecure Log (Netsecure Software), SafeSuite Decisions (Internet Security Systems), NetForensics (Cisco) et neuSecure (GuardedNet) s'appuient sur une telle approche.

4.2. *Croisement entre alertes et audits de vulnérabilité*

Une fois les traces centralisées, les fournisseurs de plates-formes ont souhaité augmenter la qualité des services offerts aux utilisateurs. Ils ont intégré le croisement *a posteriori* des alertes fournies par les systèmes de détection d'intrusions avec les rapports d'audit fournis par les analyseurs de vulnérabilité.

Ce croisement permet de savoir si la cible de l'attaquant était vulnérable à l'attaque à la date du dernier audit. C'est une approche facile à implémenter. Cela étant, elle souffre de défauts majeurs. Premièrement, elle n'apporte aucune information sur les conséquences effectives de l'attaque. En effet, la vulnérabilité peut avoir disparu depuis le dernier audit. De plus, une vulnérabilité peut être exploitée avec des conséquences différentes (voir section 3.2) et cette information n'apparaît pas dans les alertes. Deuxièmement, comme le croisement est effectué seulement de manière périodique, cette approche ne permet pas à l'opérateur de réagir en temps réel. Au moment où le croisement est effectué il y a de grandes chances que l'attaque soit achevée.

Au total, cette approche permet essentiellement de vérifier que les failles de sécurité connues ont bien été colmatées.

4.3. *Agrégation d'alertes en temps réel*

Finalement, l'approche par agrégation d'alertes a été décrite par Debar et Wespi [DEB 01] et implémentée dans le produit Tivoli RiskManager (IBM). Dans ce produit, chaque alerte est associée à un niveau de sévérité, estimant la dangerosité de celle-ci par rapport au système d'information surveillé.

Chaque alerte est décrite par un triplet constitué du nom de la signature, de l'adresse cible de l'attaque et de l'adresse source de l'attaque. A partir de cette description, une mesure de la sévérité de l'activité malveillante est effectuée en projetant chaque alerte sur les trois, deux parmi trois ou une seule des trois dimensions de la description de l'alerte. La sévérité accumulée de chacune des sept projections est pondérée par l'évolution du temps, et comparée à des seuils de référence qui indiquent si l'une des projections représente une activité dangereuse qui doit être traitée par l'opérateur.

Il s'agit ici d'une réduction du volume d'informations avec perte, et fondée uniquement sur des critères numériques. Cependant, cette agrégation fournit des résultats intéressants pour les classes d'attaques se caractérisant par de nombreux événements similaires.

5. Caractéristiques du système d'information

Pour être pertinente, la corrélation d'alertes ne doit pas se cantonner aux informations liées aux alertes et aux événements. Elle doit aussi prendre en compte les propriétés du système d'information supervisé [BRE 00].

Les caractéristiques du système d'information regroupent deux classes d'informations : la cartographie de site et la politique de sécurité. La cartographie est l'inventaire des propriétés d'un système d'information, c'est-à-dire la *topologie* (structure, éléments constitutifs, configuration des entités du réseau) et les logiciels fonctionnant sur les machines hôtes du réseau. La politique de sécurité est l'ensemble des règles qui définissent quelles actions sont autorisées ou interdites au sein du système d'information.

Dans cette section, nous présentons d'abord l'apport de ces informations pour la corrélation d'alertes, puis nous parlons de la topologie, des éléments logiciels et de la politique de sécurité. Enfin, nous évoquons un modèle de données fédérateur visant à fournir ces informations aux processus de corrélation.

5.1. *Intérêt des caractéristiques du système d'information*

Les composants physiques et logiques d'un système d'information présentent des erreurs de conception, implémentation ou configuration qui représentent des vulnérabilités. Ces vulnérabilités constituent des cibles pour les attaquants. Les cibles des attaques sont désignées sous des formes abstraites et variées dans les alertes. Il est

indispensable d'identifier la cible réelle d'une attaque. Pour ce faire, on dresse une cartographie du site supervisé qui donne un modèle du système d'information.

En la couplant avec des informations propres aux vulnérabilités connues, la cartographie permet d'estimer la menace que représente une attaque vis-à-vis des cibles réelles du système d'information en comparant les conditions de vulnérabilité avec l'état effectif de la cible.

La cartographie doit aussi permettre de connaître la visibilité topologique d'un IDS, c'est-à-dire son aptitude à détecter les manifestations d'une attaque de par sa position dans le système d'information. La visibilité topologique permet de rendre la couverture des IDS efficace (la maximiser et éviter les redondances pour limiter les doublons d'alertes). Elle permet aussi de gérer des conflits entre IDS : il est normal qu'un IDS ne réagisse pas si les manifestations d'une attaque ne lui étaient pas accessibles. Enfin, la topologie doit permettre de rendre compte de la propagation au sein d'un réseau d'attaques particulières de type ver, par exemple.

Dans [JUL 01], Julisch évoque un taux de 99 % de fausses alarmes dans certaines circonstances. Les faux positifs sont souvent abusivement attribuées aux techniques de détection simplistes utilisées dans les sondes. En réalité, la pertinence d'une alerte dépend pour une grande part de la politique de sécurité du système d'information.

5.2. *Topologie*

Au sens conventionnel du terme, la topologie désigne la topologie *physique*, c'est-à-dire la façon dont sont physiquement interconnectés les éléments du réseau d'un système d'information. Il faut toutefois bien noter que la notion de topologie s'applique à tous les niveaux des couches OSI (physique à applicative) et la modélisation correspondante diffère à chaque niveau [DOA 96].

La littérature du domaine concerne essentiellement les topologies des couches 2 et 3, à savoir respectivement la *topologie physique* (liens de données, ethernet par exemple) et la *topologie logique* (réseau, IP par exemple).

Dans le cadre de la corrélation d'alertes, les informations de topologie physique sont indispensables. En effet, la capacité d'un IDS réseau à voir des événements dépend entre autres du type d'équipements physiques utilisés dans le réseau. Les informations topologiques logiques sont elles aussi indispensables car utilisées par les IDS pour désigner les cibles et les sources des attaques.

Il faut noter que l'information sur la topologie est difficile à établir et surtout à maintenir dans des réseaux dont la taille et la complexité croissent en permanence et où les administrateurs n'ont pas toujours une maîtrise totale des éléments du réseau. Un processus automatisé, appelé *découverte topologique* est nécessaire pour alimenter le modèle à partir du réseau.

5.3. *Éléments logiciels*

Comme déjà mentionné, une majorité d'attaques exploitent des vulnérabilités liées à des erreurs d'implémentation, de configuration ou de conception des outils logiciels. La cartographie doit faire un inventaire des outils logiciels présents sur les hôtes du système d'information, c'est-à-dire recueillir les informations relatives à la nature, la version, la localisation, la configuration des outils logiciels présents dans un système d'information.

Le principal problème posé par l'inventaire logiciel réside dans l'identification précise des logiciels. Il n'existe par exemple pas de nommage canonique des versions de logiciels. Les informations relatives aux versions de logiciels vulnérables présentes dans les bases de données de vulnérabilités sont le plus souvent fantaisistes.

Un certain nombre d'outils fournissent des informations sur la présence et la configuration des outils logiciels en se basant sur des heuristiques. Nmap[1], par exemple, permet de deviner le système d'exploitation d'un hôte. En effet, le flou entourant certaines spécifications de TCP/IP donne lieu à des implémentation différentes en fonction des systèmes d'exploitation, ce qui permet à des outils comme Nmap de deviner la version d'un système d'exploitation en soumettant à l'hôte des datagrammes particuliers et en analysant la réponse.

Les outils d'audit de sécurité (scanners de vulnérabilité comme Nessus par exemple) permettent aussi d'obtenir des informations plus ou moins précises sur les applications. Les informations peuvent être directement issues d'une réponse à une requête légitime ou bien déduites d'une vulnérabilité découverte par l'outil d'audit.

Il faut malgré tout noter que d'une part ces outils ont un comportement intrusif, d'autre part ils utilisent des heuristiques plus ou moins pertinentes pour fournir des informations sur les hôtes du système d'information. Enfin, le spectre d'informations fourni par ce type d'outils est assez limité (ils ne peuvent référencer que les logiciels de type serveur). Ces outils ne suffisent pas pour l'inventaire logiciel.

Des outils d'administration réseau spécialisés dans l'inventaire logiciel sont vraisemblablement plus adéquats. L'ensemble de ces outils sont du type client/serveur : à l'image de SNMP, un client d'administration interroge des agents (serveurs) disséminés sur l'ensemble des hôtes du système d'information qui maintiennent une base de données d'informations de l'hôte concerné.

5.4. *Politique de sécurité*

Au même titre que la cartographie, la politique de sécurité contient des informations propres à un système d'information, qu'un système de corrélation d'alertes devrait prendre en compte. Une intrusion, c'est une ou plusieurs violations de la politique

1. http ://www.insecure.org/nmap/

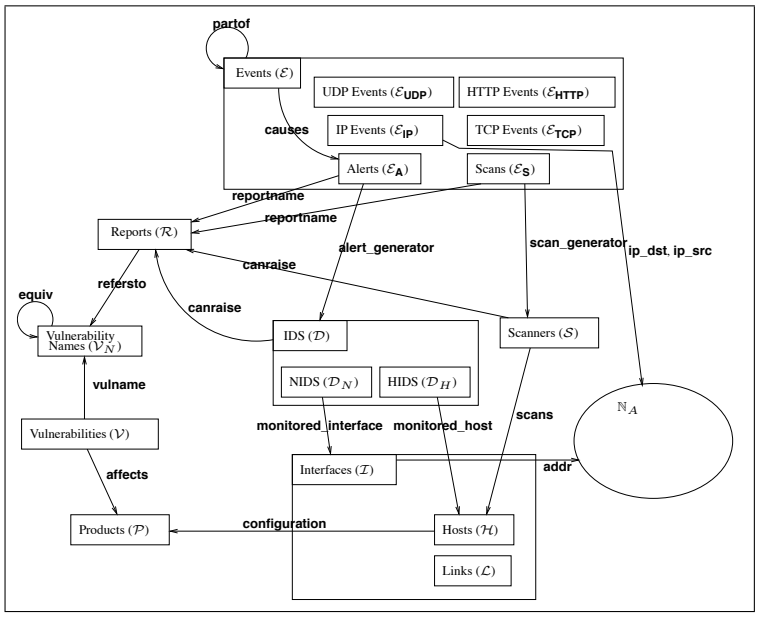

Figure 3. *M2D2 : un modèle de données utiles à la corrélation d'alertes*

de sécurité. Il serait donc intéressant de confronter le flux d'alertes à la politique, afin, par exemple, d'éliminer les alertes correspondant à des situations autorisées.

Actuellement aucun système de détection d'intrusions n'utilise d'information explicite sur la politique de sécurité pour raffiner les informations contenues dans les alertes ou pour corréler ces alertes.

5.5. *Un modèle fédérateur*

Dans [MOR 02], Morin *et al.* proposent un modèle formel des données nécessaires à la corrélation des alertes en détection d'intrusions, baptisé M2D2. Ce modèle prend pour le moment en compte quatre types d'information (les travaux actuels en corrélation ne tiennent généralement compte que du dernier type) :

1) les composants et la topologie du système d'information ;

2) les vulnérabilités connues de ces composants ;

3) les outils de sécurité passive (IDS et scanner de vulnérabilité) mis en place sur le système d'information ;

4) les actions effectuées sur le système d'information.

Certaines informations manquent à la version actuelle : les signatures d'attaque, les types de machines et les circonstances dans lesquelles ils peuvent donner lieu à alerte(s) en absence d'attaque, le comportement des sondes face à des attaques particulières, la politique de sécurité en vigueur sur le système d'information surveillé. Ces informations devraient être ajoutées dans le futur.

Ce modèle est schématisé en figure 3. Les sommets désignent les entités manipulées dans le modèle et les liens le type de relations qui les lient entre eux. Ce modèle comprend quatre grandes familles d'informations : les événements (dont font partie les alertes des IDS), les vulnérabilités, les outils de sécurité et la cartographie. Le modèle topologique utilisé dans la cartographie est inspiré de celui proposé par Vigna [VIG 96]. Les outils logiciels sont modélisés sous la forme de produits ($\mathcal{P}$). Un produit est un n-uplet contenant un nom de vendeur, un nom de produit, une version et un type. Vendeur et nom de produit sont des chaînes de caractères, la version est une suite d'entiers et le type étant un type énuméré pouvant prendre une valeur parmi plusieurs, du type : `OperatingSystem`, `HTTPServer`, `SMTPserver`, `LocalApplication`, etc.

Les vulnérabilités ($\mathcal{V}$ et **affects**), affectent une conjonction de produits. Les hôtes hébergent un ensemble de produit ($\mathcal{H}$ et **configuration**).

Ces informations permettent entre autres d'évaluer les risques associés à une attaque. Le sous-ensemble de produits vulnérables à la vulnérabilité exploitée est comparé avec le sous-ensemble de produits réellement présents sur la cible. Par exemple, une attaque exploitant une vulnérabilité propre à Apache sous Unix ne fonctionne pas contre un serveur web IIS sur Windows. Il est donc possible de mitiger les degrés de gravité des alertes émises par les IDS.

Au sein du projet IRM (*Intrusion Reference Model*), Goldman *et al.* proposent aussi un modèle de ce type [GOL 01]. Les informations cartographiques sont représentées sous forme d'une base de donnée baptisée NERD (*Network Entity Relationship Database*). NERD semble relativement complète, mais peu structurée.

6. Corrélation implicite

Contrairement à la corrélation explicite qui consiste à *reconnaître* des *scénarios* d'attaques, c'est-à-dire schémas corrélatifs prédéfinis, la corrélation *implicite* consiste à mettre en évidence des relations intrinsèques entre les alertes, sans schéma préétabli. Ces relations peuvent être des similarités entre certains attributs des alertes ou des correspondances fréquentielles ou statistiques entre des alertes. La corrélation implicite est donc une approche *non supervisée*, dans le sens où elle ne se base pas sur des schémas de corrélation appris ou spécifiés par un expert.

Les sondes de détection d'intrusions génèrent un grand nombre d'alertes. Plusieurs études, en particulier celle de Manganaris *et al.* rapportent plusieurs milliers d'alertes quotidiennes [MAN 98]. L'excès d'alertes est causé par une combinaison de phénomènes :

N	$\lvert A(N)\rvert$	$\frac{\lvert A(N)\rvert}{\lvert A(1)\rvert}$	$C(N)$	$\frac{C(N)}{C(1)}$
1	280	100 %	493427	100 %
10	99	35%	492866	99%
100	52	18%	491143	99%
500	33	11%	487318	98%
1000	26	9%	482388	97%
2000	20	7%	474053	96%
5000	13	4%	452598	91%
10000	7	2%	412661	83%
20000	3	1%	358898	72%
50000	2	<1%	337616	68%
100000	2	<1%	337616	68%
200000	1	<1%	235124	47%

Figure 4. *Illustration de la récurrence des alertes*

– les fausses alertes sont provoquées soit par des activités *légitimes*, soit par des comportements erratiques (pannes) d'entités du réseau. Dans les deux cas, ces activités sont généralement récurrentes. Elles engendrent donc un grand nombre d'alertes. Plusieurs études font état d'une proportion de 90 % de fausses alertes [LIP 00, JUL 01, JUL 02] ;

– certaines attaques se caractérisent par un grand nombre d'événements ; la granularité très fine des alertes engendre un grand nombre d'alertes ;

– la multiplication des sondes peut provoquer des redondances dans les alertes ;

– certaines sondes dissocient inutilement les alertes, même lorsqu'elles font référence au même événement.

Le tableau 4 illustre le caractère récurrent des alertes. Les mesures ont été effectuées à partir des alertes de l'IDS réseau Dragon, installé dans un réseau opérationnel, dont le trafic est important et représentatif du trafic rencontré dans un environnement de type DMZ. Dans ce tableau, $A(N)$ est l'ensemble des identifiants d'alertes dont le nombre d'occurrences est supérieur à N. $C(N)$ est l'ensemble des occurrences d'alertes dont l'identifiant est dans $A(N)$. On constate que plus de 90 % des occurrences d'alertes représentent seulement 13 identifiants d'alertes distincts.

L'objectif principal de la corrélation implicite est donc de dégager du flot d'alertes des *tendances*, en agrégeant les alertes similaires, pour limiter la quantité d'information présentée à l'opérateur de sécurité. Les mesures de similarité entre les *alertes* sont basées sur une combinaison de mesures de similarité sur leurs *attributs* et exploitent des connaissances expertes sur les attaques et les propriétés de l'environnement. Nous détaillons les principales approches dans la suite de cette section.

6.1. *Approche de Valdes et Skinner*

Dans [VAL 01], Valdes et Skinner définissent une fonction de similarité entre alertes, qu'ils utilisent pour *fusionner* des alertes similaires. Un ensemble d'alertes fusionnées est appelé *méta-alerte*. Le système est incrémental, chaque nouvelle alerte est comparée à la liste des méta-alertes existantes. Une nouvelle alerte est fusionnée avec la méta-alerte la plus proche à condition que la similarité soit jugée *suffisante*, sinon elle constitue une nouvelle méta-alerte. L'opération de fusion d'une alerte avec une méta-alerte consiste à compléter chaque attribut de la méta-alerte par l'attribut de l'alerte. Dans cette approche, la *fusion* est donc l'union ensembliste des valeurs prises par les attributs des alertes contenues dans une méta-alerte.

Les attributs retenus pour représenter les alertes sont classiquement la classe d'attaque, la source de l'attaque, la cible (machine, numéro de port), la sonde et l'estampillage temporel. La classe d'attaque est un identifiant qui représente les effets de l'attaque rapportée par l'alerte, par exemple `DenialOfService`.

La fonction de similarité entre une alerte X et une méta-alerte Y est définie comme la moyenne pondérée des mesures de similarité entre les attributs *communs* à X et Y :

$$sim(X,Y) = \frac{\sum_i e_i sim_i(X_i, Y_i)}{\sum_i e_i}$$

e_i est la pondération associée à l'attribut i ; X_i est la valeur de l'attribut i de l'alerte X. Y_i est une liste de valeurs prises par l'attribut X_i des alertes participant à la méta-alerte Y. La fonction sim_i est la fonction de similarité associée à l'attribut i, à valeur dans $[0;1]$.

Les fonctions de mesure de similarité sim_i entre attributs prennent en compte des caractéristiques propres à chaque attribut. Par exemple, deux adresses IP sources seront *similaires* si elles appartiennent au même sous-réseau. Les valeurs de similarité entre classes d'attaques sont contenues dans une matrice carrée asymétrique fournie par un expert. Deux attaques seront jugées proches si elles sont identiques, ce qui dénote plusieurs tentatives visant un même objectif ou *cohérentes*, c'est-à-dire que les effets d'une attaque permettent d'effectuer l'autre attaque. La matrice est asymétrique car la similarité entre deux classes d'attaques distinctes dépend de l'ordre d'occurence des attaques. Par exemple, comme il est plus logique pour un attaquant d'acquérir d'abord l'information sur sa cible, puis de l'attaquer, la similarité entre un *scan* de ports suivie d'un déni de service est plus importante que dans l'ordre inverse.

Les pondérations e_i utilisées dans le calcul de similarité des alertes symbolisent la *similarité attendue* des attributs. Cette pondération permet d'accroître ou au contraire de mitiger l'importance d'un attribut dans le calcul de similarité de l'alerte avec la méta-alerte. Cette pondération est fonction du contexte, qui inclut des informations liées aux autres attributs de l'alerte considérée, mais aussi des alertes précédemment impliquées dans la méta-alerte. Par exemple, dans le cas d'une attaque dont la source peut être forgée, le poids associé à la mesure de similarité de la source est faible.

En d'autres termes, sachant que la source peut prendre n'importe quelle valeur, son influence vis-à-vis de la mesure de similarité globale doit être négligeable. Dans cet exemple, l'attribut correspondant au type d'attaque a une influence sur l'attribut correspondant à la source.

En plus des pondérations utilisées pour diminuer ou accroître l'influence de certains paramètres, le calcul de similarité est contraint par des seuils de similarité minimale sur les attributs. Si la similarité entre deux attributs est inférieure au seuil minimal, alors la similarité entre l'alerte et la méta-alerte est nulle. Ces seuils permettent d'influer sur la nature des méta-alertes formées : en diminuant le seuil qui contrôle la similarité de la sonde génératrice de l'alerte, les méta-alertes peuvent fusionner des alertes issues de sondes hétérogènes ; en diminuant le seuil correspondant à la classe d'attaque, les méta-alertes peuvent contenir les alertes relatives à différentes étapes d'une attaque composée.

Les valeurs de seuils et celles utilisées dans les fonctions de similarité (en particulier la matrice de similarité des classes d'attaques) sont empiriques et fournies par un expert.

6.2. *Approche de Dain et Cunningham*

L'approche de Dain et Cunningham [DAI 01a, DAI 01b] est similaire à celle de Valdes et Skinner. Leur objectif est de former des groupes d'alertes *similaires*. L'algorithme est incrémental, les nouvelles alertes sont ajoutées au groupe le plus similaire ou font l'objet d'un nouveau *scénario*[2]. La mesure de similarité entre les alertes et les groupes d'alertes est probabiliste. Globalement, leur approche semble moins complète que celle décrite par Valdes et Skinner. En particulier, les auteurs semblent exploiter essentiellement l'attribut temporel de l'alerte et la source (adresse IP) de l'attaque.

6.3. *Approche de Cuppens*

Dans [CUP 01], Cuppens propose une technique d'*agrégation* et de *synthèse* d'alertes similaires. L'objectif est donc aussi similaire à celui de Valdes et Skinner. Une des différences entre les deux approches réside dans le fait que l'approche de Valdes et Skinner est *probabiliste*, alors que l'approche de Cuppens est basée sur des règles logiques. En d'autres termes, dans l'approche de Cuppens, deux alertes *sont* ou *ne sont pas* similaires ; l'approche de Valdes et Skinner est plus souple dans le sens où les alertes possèdent un *degré* de similarité.

La similarité des alertes est une *combinaison* de la similarité des attributs qui composent les alertes. Des règles définissant la similarité sont donc définies pour chaque type d'attribut, afin de prendre en compte leurs caractéristiques propres.

---

2. Dans ce contexte, un scénario correspond donc à une méta-alerte dans l'approche de Valdes et Skinner.

Au lieu d'utiliser une valeur qualifiant les effets d'une attaque, Cuppens utilise l'identifiant d'attaque fourni par les sondes. Comme les constructeurs d'IDS utilisent des identifiants propriétaires, la similarité de l'attribut correspondant au type d'attaque est définie à l'aide d'un référentiel commun, établi dans le cadre du projet Mirador[3]. Ce référentiel est comparable à la liste CVE[4], qui contient les identifiants de vulnérabilités connues : deux identifiants sont jugés *similaires* s'ils font référence à la même entrée dans la liste.

Des règles de similarité sont définies pour les autres attributs : source, cible, estampillage temporel.

Au cours du processus d'agrégation, les alertes sont ajoutées incrémentalement aux groupes existants, jugés similaires. Si une alerte n'est intégrable à aucun groupe, un nouveau groupe est créé, qui contient l'alerte. Notons que contrairement à l'approche de Valdes et Skinner, une alerte peut être intégrée à plusieurs groupes. Des groupes indépendants peuvent donc être eux-mêmes agrégés suite à l'intégration d'une alerte commune.

Dans l'approche de Cuppens, la *synthèse* des groupes issus de l'*agrégation* des alertes est un processus indépendant. L'agrégation consiste à créer des relations entre les alertes jugées *similaires* ; la synthèse consiste à créer une alerte *globale* pour chaque groupe d'alertes. Les attributs des alertes globales résultent de l'union (au sens ensembliste du terme) des attributs des alertes du groupe.

On peut faire un parallèle entre la *synthèse* de Cuppens qui constitue des alertes *globales* et la *fusion* de Valdes et Skinner qui constitue des *méta-alertes*. Notons toutefois que l'individualité des alertes est perdue dans le processus de fusion : on ne peut pas savoir quelle instance d'alerte présentait quels attributs. Dans l'approche de Cuppens, l'individualité des alertes est préservée ; les alertes de plus haut niveau sont d'*autres* alertes.

Les processus d'agrégation et de synthèse se situent en amont d'un autre module de corrélation, chargé de reconnaître des scénarios d'attaques (voir section 7).

6.4. *Approche de Debar et Wespi*

L'approche de corrélation de Debar et Wespi, décrite dans [DEB 01], est la première solution de corrélation d'alertes implantée dans un outil commercial, *Risk Manager*.

L'une des fonctions du *composant d'agrégation et de corrélation* (ACC) de *Risk Manager* est de former des groupes d'alertes similaires, appelés *situations*.

Les alertes manipulées sont des triplets constitués d'un identifiant d'attaque, de la source et de la cible de l'attaque. Une *situation* est un ensemble d'alertes ayant la

3. Programme d'Étude Amont de la DGA, 1999-2002.
4. http ://cve.mitre.org

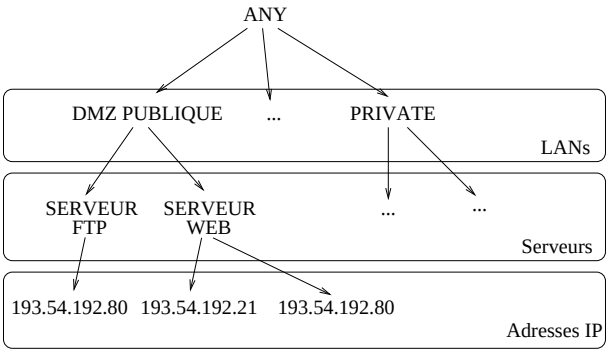

Figure 5. *Hiérarchie pour les cibles des attaques*

même projection selon un *certain nombre d'axes*, les axes étant représentés par les attributs.

Le nombre et la nature des axes utilisés pour la projection ont une signification particulière. Par exemple, des alertes ayant la même source et le même identifiant peuvent révéler un attaquant tentant d'exploiter une même vulnérabilité, indépendamment de la cible ; des alertes ayant la même source et la même cible peuvent être révélatrices d'un attaquant intéressé par un hôte donné (un serveur web par exemple).

6.5. *Approche de Julisch*

Dans [JUL 01, JUL 02], Julisch propose d'adapter une méthode de fouille de données connue sous le nom d'AOI (*Attribute-Oriented Induction*) pour grouper les alertes et identifier le phénomène à l'origine des groupes d'alertes.

De manière générale, l'AOI consiste à fusionner des données représentées par des $n-$uplets d'attributs en fonction de *hiérarchies de concepts* (ou *taxonomies*), liées à chaque attribut.

Formellement, une hiérarchie de concepts est un ensemble fini constitué de l'ensemble des valeurs que peut prendre un attribut, muni d'un ordre partiel. Les hiérarchies de concept sont classiquement représentées par un diagramme de Hasse. Le niveau d'abstraction des valeurs d'attributs (ou nœuds) des diagrammes va croissant des feuilles au sommet de l'arborescence. Les attributs des données de base – en l'occurence les alertes issues des sondes – appartiennent aux feuilles des arborescences.

La figure 5 représente la hiérarchie de l'attribut correspondant à la cible des attaques. Le niveau d'abstraction le plus bas est composé d'adresses IP (contenues dans les alertes), suivi de types d'hôtes particuliers, puis de domaines du réseau surveillé.

Dans l'approche de Julisch, les alertes sont des quadruplets $(ident, source, cible, t)$ où $ident$ est l'identifiant de l'attaque fourni par l'IDS, $source$ est l'adresse IP source de l'attaque, $dest$ l'adresse IP destination et t la date d'occurrence.

Le mécanisme d'AOI consiste à abstraire progressivement les valeurs des attributs des données en suivant la hiérarchie de concepts associée à chaque type d'attribut et à fusionner les données rendues égales par l'abstraction. De cette manière, le nombre de tuples est diminué et les valeurs des attributs sont plus abstraites. Les alertes fournies à l'opérateur sont moins nombreuses et moins spécifiques. L'un des avantages de cette approche est d'intégrer des connaissances environnementales dans les arbres de concepts (la topologie, par exemple) et de les exploiter pour *qualifier* les groupes d'alertes : les attributs des alertes de haut niveau fournies à l'opérateur ne sont pas l'union des attributs des alertes du groupe.

L'algorithme d'AOI conventionel consiste à abstraire les valeurs des attributs jusqu'à ce que le nombre total de données devienne inférieur à un seuil fixé. Ce critère d'arrêt est mal adapté à la corrélation d'alertes car le nombre souhaitable d'alertes issues du processus d'abstraction n'est pas connu au préalable. De plus, dans le processus d'AOI conventionnel, une abstraction d'alerte selon un attribut est faite sur l'ensemble des données, même si cela n'est pas pertinent. Cette abstraction sauvage conduit à une surgénéralisation des données.

Pour ces raisons, Julisch propose un algorithme modifié d'AOI. A chaque itération de l'algorithme, un attribut est selectionné selon une heuristique pour être généralisé. Les groupes d'alertes qui se généralisent en une même alerte et contenant plus d'alertes qu'un seuil préalablement fixé sont retirés de l'ensemble des alertes à généraliser et sont fournis à l'opérateur. L'abstraction effectuée sur les autres alertes est annulée et un autre attribut est choisi pour être généralisé. De cette manière, les alertes ne sont pas surgénéralisées.

Cette approche permet par exemple de constituer des alertes du type « un **serveur proxy** génère des **scans de ports** vers l'**extérieur** » (les termes en gras sont les attributs des alertes générées par le mécanisme de corrélation). **Serveur proxy** est la source issue de l'abstraction de l'adresse IP d'un serveur proxy ; **scan de port** est un identifiant d'attaque n'ayant subi aucune abstraction ; **extérieur** est la destination de l'attaque, issu de plusieurs abstractions de la cible.

L'approche de Julisch n'a pas pour objectif de construire des scénarios d'attaques, mais plutôt d'effectuer des regroupements d'alertes correspondant à des *tendances* remarquables dans une base d'alertes. L'opérateur peut traiter les alertes par lots et donc se concentrer sur les alertes éventuellement plus sévères.

L'approche de Julisch a toutefois quelques inconvénients. Tout d'abord, les hiérarchies des structures d'attributs sont des arbres, par conséquent leur pouvoir expressif est limité. Des graphes acycliques seraient préférables. Ensuite, le système d'AOI n'est pas incrémental : il doit être réexécuté à chaque fois que l'opérateur souhaite avoir un condensé des alertes. Enfin, l'ordre de généralisation des attributs est basé sur des heuristiques qui sont discutables.

6.6. *Synthèse sur les approches implicites*

La corrélation implicite n'utilise pas de schéma corrélatif préétabli. Elle exploite les informations intrinsèques des alertes pour identifier des relations entre les alertes, les regrouper et synthétiser l'information. Le contenu des alertes fournies à l'opérateur est plus important et le nombre global d'alertes est diminué.

Le vocabulaire employé est assez varié mais recouvre des notions et des fonctionnalités très similaires. L'ensemble des approches de corrélation implicite repose sur une définition de similarité entre les alertes. La mesure de similarité permet d'*agréger* les alertes, c'est-à-dire effectuer des regroupements d'alertes jugées plus ou moins proches en fonction de leurs attributs. Les fonctions de similarité entre les alertes sont une combinaison de fonctions de similarité définies sur leurs attributs. Les fonctions de similarité sur les attributs sont basées sur des connaissances expertes liées aux attaques et à l'environnement.

La *synthèse* permet de constituer des alertes de haut niveau, dont les attributs condensent ceux des alertes agrégées. Cette synthèse consiste généralement à faire l'union, au sens ensembliste du terme, des valeurs des attributs, ce qui est assez limité.

Notons que les approches implicites bénéficieraient d'un consensus sur un langage permettant de *décrire* les alertes émises par les sondes ; seul un identifiant d'attaque est généralement disponible à l'heure actuelle dans les alertes pour *décrire* le type d'attaque, ce qui ne permet pas de traiter cet attribut de manière automatisée. Par exemple, les informations concernant les prérequis et les effets d'une attaque permettent de rapprocher des attaques qui constituent un enchaînement logique ; certaines classes d'attaques telles que les dénis de services invalident la source des attaques en tant qu'attribut corrélatif. Ces informations sont disponibles, mais ne sont pas formalisées. Plusieurs tentatives de formalisation ont été effectuées, parmis lesquelles [KEN 99, GOL 01, HOW 98], mais aucune n'a fait l'objet d'un consensus parmi les acteurs du domaine.

7. Corrélation explicite

L'objectif que l'on poursuit avec la corrélation explicite est la détection de scénarios d'attaque complexes, c'est-à-dire impliquant un enchaînement d'actions de la part de l'attaquant. Les sondes en place sur le système attaqué émettent des alertes pour tout ou partie des actions d'un attaquant. Pour détecter l'ensemble de l'attaque, il faut donc être capable de corréler ces alertes.

A cette fin, l'idée de base consiste à confronter le flux d'alertes à des scénarios d'attaques connus *a priori*. La corrélation explicite est donc à rapprocher de l'approche par scénario, classique en détection d'intrusions. Cependant, elle s'en distingue en utilisant des signatures plus évoluées.

Le domaine de la corrélation explicite a été plus largement défriché que celui de la corrélation implicite. Cependant, la corrélation explicite est généralement appli-

quée directement aux événements à l'intérieur de sonde (voir, par exemple, ASAX [HAB 93], STAT [VIG 99], BMSL [UPP 01], QuickSand [KRÜ 01], Sutekh [POU 01], LogWeaver [ROG 01]) et non au flux d'alertes, même si la référence [TOT 04] propose un langage de corrélation pouvant s'appliquer aux événements ou aux alertes. Au final, les travaux portant sur la corrélation explicite d'alertes restent peu nombreux.

Dans la suite de cette section nous présentons deux approches de corrélation explicite d'alertes. Nous commençons par détailler une modélisation de scénarios par programmation logique[5] (section 7.1), puis, nous présentons plus brièvement une spécification des scénarios par chroniques (section 7.2). Enfin, la section 7.3 conclut cette partie par une courte synthèse.

7.1. *Modèles de scénarios par programmation logique*

L'approche adoptée dans [CUP 02b] suppose que l'attaquant dispose d'un ensemble d'attaques élémentaires lui permettant d'atteindre son objectif d'intrusion à partir d'un état initial qui est celui du système avant le début du scénario. Les attaques élémentaires sont représentées par leur nom, leur préconditions et leur postconditions. Les préconditions d'une attaque sont les conditions qui doivent être satisfaites pour que l'attaque réussisse et les postconditions sont les effets de l'attaque sur l'état du système. Ces conditions sont représentées par des expressions logiques du premier ordre. Le formalisme présenté ne permet pas d'inclure des disjonctions dans les postconditions des attaques. La conséquence d'une attaque doit donc pouvoir être prédite de manière déterministe. Avant d'expliciter les mécanismes de la corrélation d'alertes nous allons introduire les différents types de corrélation que nous pouvons rencontrer.

7.1.1. *Différents types de corrélation explicite*

Le principe de l'approche envisagée dans [CUP 02b, CUP 02c] pour la fonction de corrélation est d'analyser la spécification des attaques en LAMBDA [CUP 00] pour reconnaître automatiquement des liens logiques entre les attaques. Deux types principaux de liens logiques sont considérés :

– lien de cause à effet ou corrélation *post-pré*. Il s'agit d'établir un lien logique entre la postcondition d'une première attaque A_1 et la précondition d'une seconde attaque A_2. Ce lien est de type *cause à effet* : la réalisation de l'attaque A_1 a rendu possible (ou à contribuer à rendre possible) la réalisation de l'attaque A_2. Par conséquent, si l'on observe une occurrence de A_1 puis une occurrence de A_2, on peut corréler ces deux attaques. On simule ainsi un attaquant qui réalise A_1 dans le but de pouvoir ensuite réaliser A_2 ;

– corrélation sur les buts ou corrélation *post-post*. Il s'agit d'établir un lien logique entre les postconditions de deux attaques A_1 et A_2. Ce lien est de type *corrélation*

5. Pour un exemple de modélisation de scénarios par programmation impérative, voir [MIC 01, TOT 04].

sur les buts : la réalisation des attaques A_1 et A_2 vise à atteindre le même but. Par conséquent, si l'on observe une occurrence de A_1 et une occurrence de A_2 (il n'y a pas nécessairement de lien temporel entre A_1 et A_2), alors on peut corréler ces deux attaques. Remarque : l'une des attaques (voire les deux attaques) a peut-être échoué. Dans ce cas, en corrélant les deux attaques, on simule un attaquant qui veut atteindre un certain objectif d'intrusion et qui essaye pour cela différentes attaques qui permettent (si elles réussissent) d'atteindre cet objectif.

7.1.1.1. Plan d'intrusion

Le comportement d'un attaquant réalisant une intrusion dans un système informatique est représenté par un *plan d'intrusion*. Un plan d'intrusion est composé d'un ensemble d'actions permettant à l'attaquant de progresser dans le système informatique : ces actions sont les attaques, correspondant à l'exploitation d'une vulnérabilité du système informatique. Certaines attaques peuvent être corrélées, par exemple quand la réalisation d'une attaque permet d'en effectuer une deuxième. Chaque plan correspond à un ensemble d'attaques cohérentes qui peuvent être corrélées. A un moment donné du déroulement de l'intrusion, ce plan est partiellement réalisé par l'attaquant. Le module de corrélation doit donc générer l'ensemble des différents plans potentiels que peut suivre un attaquant, compte tenu des descriptions de vulnérabilités dont il dispose et des attaques précédemment identifiées. Au fur et à mesure, les plans potentiels sont mis à jour grâce aux informations fournies par les différents modules de détection d'intrusions.

7.1.1.2. Corrélation directe

Dans la suite de ce document, les attaques sont représentées sous une forme faisant appel aux notions de pré et postconditions.

$Pre \Longrightarrow^{A} Post$: si la proposition *Pre* est satisfaite dans un état initial e_i alors l'exécution de l'action A conduit à un état final e_f dans lequel la proposition *Post* est satisfaite.

La corrélation de deux actions $Pre_1 \Longrightarrow^{A_1} Post_1$ et $Pre_2 \Longrightarrow^{A_2} Post_2$ s'appuie sur les liens logiques pouvant exister entre les pré et postconditions ($Post_1$ et Pre_2 par exemple). En particulier, on dit que les actions A_1 et A_2 sont directement corrélées si et seulement si on a simultanément : $Pre_1 \Longrightarrow^{A_1} Post_1$, $Pre_2 \Longrightarrow^{A_2} Post_2$ et $Post_1 \rightarrow Pre_2$ où $\rightarrow$ est une implication logique classique (implication matérielle). Dans la suite le fait « A_1 et A_2 sont directement corrélées » sera noté $correlation\_directe(A_1, A_2)$. Cependant, une simple relation causale ($Post_1 \rightarrow Pre_2$) n'est pas suffisante pour définir correctement la corrélation entre deux actions. En effet, une telle définition est trop restrictive pour prendre en compte toutes les possibilités de corrélation. C'est la raison pour laquelle la notion de corrélation conditionnelle est introduite.

7.1.1.3. Corrélation conditionnelle

A_1 et A_2 sont corrélées conditionnellement (sous l'hypothèse h) si et seulement si on a simultanément $Pre_1 \Longrightarrow^{A_1} Post_1$, $Pre_2 \Longrightarrow^{A_2} Post_2$ et $Abduction(Post_1, Pre_2, h)$.

$Abduction(Post_1, Pre_2, h)$ signifie qu'il est possible d'utiliser $Post_1$ pour déduire Pre_2 à condition de faire l'hypothèse h. $Abduction(Post_1, Pre_2, h)$ est défini de la façon suivante :

– $Post_1 \wedge h \rightarrow Pre_2$ (la postcondition de A_1 et l'hypothèse h impliquent la précondition de A_2) ;

– on n'a pas $h \rightarrow Pre_2$ (l'hypothèse h seule n'implique pas la précondition de A_2) ;

– on n'a pas $\neg(h \wedge Post_1)$ (la postcondition de A_1 n'est pas inconsistante avec l'hypothèse h).

Dans la suite le fait « A_1 et A_2 sont corrélées conditionnellement sous l'hypothèse h » sera noté $correlation\_conditionnelle(A_1, A_2, h)$. Remarquons que la corrélation directe est un cas particulier de la corrélation conditionnelle :

$correlation\_directe(A_1, A_2) \leftrightarrow correlation\_conditionnelle(A_1, A_2, h) \wedge h = Vrai$

7.1.1.4. Corrélation chaînée

Les définitions ci-dessus de corrélation directe et de corrélation conditionnelle sont étendues pour prendre en compte le cas où il existe une chaîne de corrélations entre deux attaques A_1 et A_n. Il existe une corrélation chaînée conditionnelle entre A_1 et A_n si et seulement si : $\exists A_2, \ldots, A_{n-1}$ tels que , $\forall i \in [1, ...n-1]$, A_i et A_{i+1} sont corrélées conditionnellement sous l'hypothèse $h^{(i+1)}$. Si toutes les corrélations entre A_i et A_{i+1} sont des corrélations directes alors on dira qu'il existe une corrélation chaînée directe entre A_1 et A_n.

7.1.1.5. Corrélation sur les objectifs

Nous avons fait l'hypothèse qu'un attaquant se sera généralement fixé un but avant de commencer son scénario d'attaque, par exemple l'obtention des droits root sur une machine ou rendre une machine indisponible (un serveur web ou un DNS par exemple). Ces objectifs d'intrusion peuvent s'exprimer par une expression logique sur l'état du système et correspondent à une violation de la politique de sécurité. Prenons l'exemple d'un objectif d'intrusion qui consiste à rendre indisponible une machine DNS. Cet objectif d'intrusion peut être modélisé de la manière suivante :

Objectif_intrusion : $DOS\_sur\_DNS(Host)$

Conditions : $serveur\_dns(Host) \wedge dos(Host)$

$serveur\_dns(Host)$ signifie que la machine $Host$ est un serveur de nom de domaine et $dos(Host)$ signifie que la machine $Host$ est indisponible à cause d'une attaque DOS. Une action permet d'atteindre un objectif d'intrusion si l'on peut déduire de sa postcondition les conditions d'un objectif d'intrusion.

7.1.2. *Schémas de base de la corrélation*

Cette section récapitule les principes de base de l'approche pour spécifier la fonction de corrélation d'alertes. La fonction de corrélation prend en entrée des alertes générées par diverses sondes. Lorsqu'une nouvelle alerte $Alert_i$ arrive en entrée de la fonction de corrélation, cette fonction recherche avec quelles autres alertes déjà présentes, $Alert_i$ peut être corrélée. Pour cela, il faut consulter la description de $Alert_i$ pour déterminer l'attaque $Attack_i$ associée à cette alerte. On recherche ensuite les faits de la forme $correlation\_conditionnelle(Attack_j, Attack_i, Cond)$ ou de la forme $correlation\_conditionnelle(Attack_i, Attack_k, Cond)$.

S'il existe de tels faits et si des alertes de la base correspondent à des attaques de type $Attack_j$ ou $Attack_k$, alors on insère l'alerte $Alert_i$ dans le plan en cours de construction. Dans chaque cas, il faudra auparavant vérifier les conditions associées aux corrélations. On peut naturellement étendre la démarche en remplaçant la corrélation conditionnelle par une corrélation chaînée. Les schémas restent identiques sauf qu'on introduit alors dans le plan, entre $Attack_i$ et $Attack_j$, des attaques qui n'ont été détectées par aucun module de détection d'intrusions. L'objectif de ce type de corrélation est double. Elle permet d'une part de détecter des attaques coordonnées cherchant toutes à atteindre un même but. Elle peut d'autre part être utilisée pour détecter des tentatives d'attaques visant le même but qui sont réitérées jusqu'à ce que l'une d'elles réussisse.

7.1.3. *Génération des règles de corrélation*

Une règle permet au processus de corrélation d'établir la relation entre deux alertes correspondant aux attaques corrélées. Pour cela une base d'attaque décrite en LAMBDA doit être préalablement construite. L'objectif de la compilation de la base d'attaques est de générer les triplets de la forme $(A, B, Cond)$ où A et B sont deux attaques décrites en LAMBDA et Cond est une expression logique. Si l'algorithme de compilation permet de générer un tel triplet, alors on insérera dans la base gérée par Prolog, le fait $correlation\_conditionnelle(A, B, Cond)$.

7.1.4. *Génération des plans en cours d'exécution*

Supposons que la fonction de corrélation reçoive deux alertes $Alert_1$ et $Alert_2$. En consultant le champ *classification* de ces deux alertes, on peut extraire le type de l'attaque $Attack_1$ et $Attack_2$ associé respectivement à $Alert_1$ et $Alert_2$. On peut alors chercher dans la base s'il existe un fait de la forme :

$$correlation\_alert(Attack_1, Attack_2, Cond)$$

Pour que la corrélation de $Attack_1$ et $Attack_2$ soit envisageable, il faut ensuite vérifier que l'alerte $Alert_1$ a eu lieu avant $Alert_2$. Pour cela, il suffit de tester le champ ***detecttime*** de $Alert_1$ et $Alert_2$. Il reste alors à vérifier que la condition de corrélation $Cond$ est satisfaite pour pouvoir corréler les $Alert_1$ et $Alert_2$.

Pour cela, une partie des informations sera en général disponible dans la description des alertes $Alert_1$ et $Alert_2$. Par exemple, Cond peut imposer que les sources des alertes $Alert_1$ et $Alert_2$ soient identiques pour que la corrélation puisse avoir lieu. Ce type d'informations est disponible en consultant la description des alertes $Alert_1$ et $Alert_2$. La description des alertes doit donc permettre de vérifier partiellement si Cond est satisfaite. Lorsque Cond ne peut être totalement vérifié en consultant la description des alertes, les informations manquantes sur l'état du système seront recherchées dans la base de données M2D2.

7.1.5. *Anticorrélation*

Nous évoquons succinctement dans cette partie la notion d'anticorrélation qui se déduit aisément de la notion de corrélation. Nous prendrons le cas de l'anticorrélation directe entre une postcondition et une précondition, le cas de l'anticorrélation indirecte s'en déduisant facilement. On dit que les actions A_1 et A_2 sont directement anticorrélées si on a : $\neg Post_1 \Longrightarrow Pre_2$ ou $Post_1 \Longrightarrow \neg Pre_2$.

Cette notion permet de dire qu'une séquence d'attaques ne réussira pas et qu'elle ne correspond pas à un scénario d'attaque. Grâce à ce raisonnement il devient possible de réduire le nombre de faux positifs sur les scénarios. Une autre application de la notion d'anticorrélation serait d'intégrer, dans l'analyse, des actions qui empêcheraient l'attaquant d'atteindre l'objectif d'intrusion qu'il s'est fixé. Ceci est particulièrement utile pour prendre en compte, dans le processus de corrélation, les effets d'un processus de réaction aux alertes. Cette approche est envisagée dans [CUP 02a].

7.2. *Spécification des scénarios comme chroniques*

Un système de reconnaissance de chroniques donne une interprétation de l'évolution du monde, étant donné des événements datés. Ce système prend en entrée un flux d'événements datés et reconnaît des instances de chroniques au fur et à mesure qu'elles se développent. Chaque chronique peut être vue comme un ensemble de motifs événementiels sur lesquels un ensemble de contraintes contextuelles et temporelles s'appliquent. Si les événements observés correspondent aux motifs de la chronique et si leur occurrence ne viole pas les contraintes temporelles et contextuelles, alors une instance de la chronique est reconnue.

Le système de chroniques de Dousson [DOU 94] s'appuie sur une prévision complète des dates possibles pour chaque événement encore attendu ; l'ensemble de ces valeurs (appelées *fenêtres temporelles*) est réduit grâce à la propagation des dates d'événements observés au travers du graphe de contraintes temporelles de la chro-

nique. La reconnaissance est incrémentale – chaque événement est intégré dès son arrivée – et en une seule lecture du flot d'entrée.

Un tel système est donc bien adapté à la détection d'intrusions. Chaque alerte constitue un événement daté et chaque attaque peut être décrite par une chronique. En outre, on peut aussi décrire sous forme de chronique des phénomènes récurrents ou donnant naissance à des alertes redondantes, ce qui permet un regroupement des alertes correspondantes.

Ainsi, les chroniques ont été utilisées dans le cadre du projet Mirador pour corréler des alertes issues de sondes de détection d'intrusions, sur la base de scénarios d'attaque prédéfinis.

Pour autant, les chroniques ont été initialement conçues pour la supervision de réseaux de télécommunications (détection de pannes de composants réseau). Les chroniques représentent alors des phénomènes de pannes. Elles sont écrites par des experts capables de décrire les manifestations d'une panne en termes de types d'événements et de délais. Les pannes sont des phénomènes récurrents et relativement déterministes. Un nombre relativement faible de chroniques permet donc de décrire la quasi-totalité des phénomènes de pannes. En revanche, dans le domaine de la détection d'intrusions, le déterminisme des scénarios employés par les attaquants est discutable, non seulement en termes d'alertes constatées, mais aussi en termes de délais interattaques. L'exhaustivité de la base de chroniques censées représenter les scénarios d'attaques possibles est donc difficile à assurer. D'autre part, les scénarios d'attaques sont rarement des phénomènes récurrents. L'élaboration de chroniques censées les représenter est donc limitée. Enfin, les étapes d'un scénario d'attaque peuvent être dans certains cas indétectables et ne peuvent, par conséquent, pas faire partie des chroniques.

Toutefois, les flots d'alertes générés par les outils de détection d'intrusions présentent indéniablement des phénomènes récurrents, qui ne sont pas pour autant des *scénarios d'attaques*. Ces phénomènes peuvent être d'origine malveillante ou non. Ils impliquent plusieurs alertes et par conséquent l'utilisation de chroniques est intéressante. Les attaques de type ver (Nimda, par exemple) sont un exemple de phénomènes récurrents qui impliquent beaucoup d'alertes : dans le cas de Snort, une seule tentative d'attaque par Nimda provoque plus de 15 alertes. La référence [MOR 03] présente ce type d'utilisation des chroniques.

D'une manière générale, les chroniques peuvent à la fois représenter des phénomènes *anormaux* et des phénomènes *normaux*. En détection d'intrusions, il est donc envisageable d'utiliser les chroniques pour discriminer les faux positifs des vrais positifs en impliquant des événements anodins dans le processus de corrélation afin de consolider les alertes. Ces types d'applications de chroniques sont actuellement en cours d'étude.

7.3. *Synthèse sur la corrélation explicite*

La corrélation explicite permet de regrouper des alertes générées par une même attaque ou par les étapes élémentaires d'une attaque de plus grande ampleur. A cette fin, le flux d'alertes est confronté à des signatures d'attaques connues *a priori* et stockées dans une base. Ces signatures sont exprimées dans des langages évolués, dont nous avons donné deux exemples dans cet article.

Si de nombreux travaux traitent de corrélation d'événements, peu se sont intéressé à la corrélation d'alertes. Les travaux de Cuppens, détaillés ci-dessus, comptent parmi les plus avancés. Ces travaux s'appuient sur une description des attaques dans le langage LAMBDA. Ils sont donc confrontés au même problème que l'ensemble des travaux traitant de détection par une approche à base de scénarios : la constitution et, surtout, la maintenance de la base de signatures.

La corrélation explicite paraît en outre un bon outil pour traiter des phénomènes d'alertes récurrentes (cas des vers) ou redondantes (lorsque plusieurs sondes détectent la même attaque, par exemple). Pour ces types de phénomène, les chroniques semblent constituer un outil particulièrement bien adapté.

8. Conclusion

Cet article de synthèse[6] a proposé un état de l'art des techniques de corrélation d'alertes utilisées dans le domaine de la détection d'intrusions.

Nous avons défini un vocabulaire commun qui permet d'harmoniser les différents travaux réalisés dans le domaine. Nous avons également défini les objectifs que peut avoir la corrélation d'alertes et proposé trois familles de techniques permettant de construire des algorithmes de corrélation permettant de satisfaire un ou plusieurs de ces objectifs.

La distinction de premier niveau que nous faisons entre les approches de corrélation présentées dans cet article s'appuie sur les connaissances expertes nécessaires à la mise en œuvre de l'approche. Dans le cas de la corrélation implicite, l'expertise porte sur les similarités entre les valeurs d'attributs des alertes ; dans le cas de la corrélation explicite, l'expertise porte sur les modalités d'attaque.

Chacune de ces deux familles de corrélation permet de réduire le trop grand nombre d'alertes produit actuellement par les outils de détection d'intrusions. En outre, les informations portées par les groupes d'alertes constitués par la corrélation sont moins parcellaires et de plus haut niveau que les informations portées initialement par chacune des alertes. La corrélation permet donc d'atteindre les deux objectifs fixés et nécessaires selon nous à l'évolution des systèmes de détection d'intrusions : la réduc-

6. Ce travail a été en partie réalisé dans le cadre du projet RNTL (Réseau National des Technologies Logicielles) DICO (Détection d'Intrusions COopérative). Les auteurs remercient le RNTL et le ministère de la Recherche pour leur soutien.

tion du volume d'informations à traiter par l'opérateur humain et l'amélioration de la qualité du diagnostic fourni à cet opérateur.

A notre connaissance, il n'existe pas à l'heure actuelle de travaux combinant l'approche explicite et l'approche implicite. Pour autant, les travaux conduits actuellement autour de l'exploitation du modèle M2D2 poursuivent en partie cet objectif, les informations de M2D2 étant exploitées *via* l'une et l'autre de ces approches. Il reste cependant à étudier la corrélation des alertes synthétiques issues des deux approches.

Restent à éprouver les approches proposées en environnement opérationnel. Dans un tel cadre, avec un flux d'alertes important et de nombreux types d'attaque possibles, les performances des algorithmes de corrélation sont essentielles. Actuellement, nous n'avons pas connaissance de travaux comparant les performances des divers mécanismes proposés. C'est donc un travail très important qui reste à faire. En outre, toujours en environnement opérationnel, la simplicité et la stabilité de la configuration des mécanismes de corrélation est un autre élément déterminant. Ce point est particulièrement critique pour la corrélation explicite, les administrateurs de sécurité ne pouvant passer leur temps à mettre à jour la base de signatures d'attaque. Il nous semble qu'une solution passe par l'adoption par l'ensemble de la communauté d'un langage d'expression de corrélation d'alertes. En effet, avec un langage commun, on peut envisager une mutualisation des signatures, comme celle que l'on connaît dans le cadre du langage de filtrage d'événements de SNORT, par exemple. Des langages comme LAMBDA ou ADeLe [MIC 01, TOT 04] poursuivent cette ambition.

Prometteuse en termes de mécanismes proposés, la corrélation d'alertes doit donc encore faire ses preuves en termes de passage à l'échelle.

9. Bibliographie

[BRE 00] BREITBART Y., GAROFALAKIS M. N., MARTIN C., RASTOGI R., SESHADRI S., SILBERSCHATZ A., « Topology Discovery in Heterogeneous IP Networks », *INFOCOM (1)*, 2000, p. 265-274.

[CUP 00] CUPPENS F., ORTALO R., « LAMBDA : A Language to Model a Database for Detection of Attacks », DEBAR H., MÉ L., WU S. F., Eds., *Proceedings of the Third International Workshop on the Recent Advances in Intrusion Detection (RAID'2000)*, LNCS 1907, October 2000, p. 197-216.

[CUP 01] CUPPENS F., « Managing Alerts in Multi-Intrusion Detection Environment », *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC'01)*, 2001.

[CUP 02a] CUPPENS F., AUTREL F., MIÈGE A., BENFERHAT S., « Correlation in a Intrusion Detection Process », *SEcurity of Communications on the Internet (SECI 2002)*, Tunis, Tunisie, September 2002.

[CUP 02b] CUPPENS F., AUTREL F., MIÈGE A., BENFERHAT S., « Recognizing Malicious Intention in an Intrusion Detection Process », *Second International Conference on Hybrid Intelligent Systems*, Santiago, Chili, December 2002, Special session "Hybrid Intelligent Systems for Intrusion Detection".

[CUP 02c] CUPPENS F., MIÈGE A., « Alert correlation in a cooperative intrusion detection framework », *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, CA, May 2002.

[DAI 01a] DAIN O., CUNNINGHAM R., « Building Scenarios from a Heterogeneous Alert Stream », *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, June 2001.

[DAI 01b] DAIN O., CUNNINGHAM R., « Fusing a heterogeneous alert stream into scenarios », *Proceedings of the 2001 ACM Workshop on Data Mining for Security Applications*, November 2001, p. 1–13.

[DEB 01] DEBAR H., WESPI A., « Aggregation and Correlation of Intrusion-Detection Alerts », LEE W., MÉ L., WESPI A., Eds., *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID 2001)*, LNCS 2212, Davis, CA, USA, October 2001, Springer, p. 85-103.

[DEB 02] DEBAR H., MORIN B., « Evaluation of the diagnostic capabilities of commercial intrusion desctection systems », ET AL. A. W., Ed., *Proceedings of the fifth international Symposium on Recent Advances in Intrusion Detection (RAID 2002)*, Springer, LNCS 2516, 2002, p. 177-198.

[DOA 96] DOAR M., « A Better Model for Generating Test Networks », *Proceedings of Globecom'96*, November 1996.

[DOU 94] DOUSSON C., « Suivi d'Evolutions et Reconnaissance de Chroniques », PhD thesis, Université Paul Sabatier, Toulouse, 1994.

[GOL 01] GOLDMAN R. P., HEIMERDINGER W., HARP S. A., GEIB C. W., THOMAS V., CARTER R. L., « Information Modeling for Intrusion Report Aggregation », *Proceedings of the DARPA Information Survivability Conference and Exposition*, June 2001.

[HAB 93] HABRA N., CHARLIER B. L., MOUNJI A., MATHIEU I., « ASAX : Software architecture and rule-based language for universal audit trail analysis », *Proc of European Symposium on Research in Computer Security*, Toulouse, 1993.

[HOW 98] HOWARD J. D., LONGSTAFF T. A., « A common language for computer security incidents », CERT - SAND98-8667, http ://www.cert.org/research/taxonomy_988 667.pdf, 1998.

[JUL 01] JULISCH K., « Mining Alarm Clusters to Improve Alarm Handling Efficiency », *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC)*, December 2001.

[JUL 02] JULISCH K., DACIER M., « Mining Intrusion Detection Alarms for Actionable Knowledge », *Proceedings of Knowledge Discovery in Data and Data Mining (SIGKDD)*, 2002.

[KEN 99] KENDALL K., « A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems », Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1999.

[KRÜ 01] KRÜGEL C., TOTH T., KERER C., « Decentralized Event Correlation for Intrusion Detection », VERLAG S., Ed., *Proceedings ot the 4th International Conference on Information Security and Cryptology (ICISC 2001)*, Lecture Notes in Computer Science, December 2001.

[LIP 00] LIPPMANN R. P., FRIED D. J., GRAF I., HAINES J. W., KENDALL K. R., MCCLUNG D., WEBER D., WEBSTER S. E., WYSCHOGROD D., CUNNINGHAM R. K., ZISS-

MAN M. A., « Evaluating Intrusion Detection Systems : The 1998 DARPA Off-line Intrusion Detection Evaluation », *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX'00)*, 2000.

[MAN 98] MANGANARIS S., CHRISTENSEN M., ZERKLE D., HERMIZ K., « A Data Mining Analysis of RTID Alarms », *First International Workshop on the Recent Advances in Intrusion Detection (RAID98)*, September 1998.

[MIC 01] MICHEL C., MÉ L., « ADeLe : an Attack Description Language for Knowledge-based Intrusion Detection », *Proceedings of the 16th International Conference on Information Security*, Paris, France, June 2001, Kluwer, p. 353-365.

[MOR 02] MORIN B., MÉ L., DEBAR H., DUCASSÉ M., « M2D2 : a formal data model for IDS Alert Correlation », *Proceedings of the fifth international Symposium on Recent Advances in Intrusion Detection (RAID 2002)*, Springer, LNCS 2516, 2002, p. 97-104.

[MOR 03] MORIN B., DEBAR H., « Correlation of Intrusion Symptoms : an Application of Chronicles », *Proceedings of the sixth International Symposium on the Recent Advances in Intrusion Detection (RAID'2003)*, October 2003.

[POU 01] POUZOL J.-P., DUCASSÉ M., « From Declarative Signatures to Misuse IDS », LEE W., MÉ L., WESPI A., Eds., *Recent Advances in Intrusion Detection, Proceedings of the 4th International Symposium*, LNCS 2212, 2001, p. 1-21.

[ROG 01] ROGER M., GOUBAULT-LARRECQ J., « Log Auditing through Model-Checking », *Proceeding of the 14th Computer Security Foundations Workshop*, IEEE Computer Society Press, 2001.

[TOT 04] TOTEL E., VIVINIS B., MÉ L., « A Language Driven Intrusion Detection System for Event and Alert Correlation », *Proceedings ot the 19th IFIP International Information Security Conference*, Kluwer Academic, August 2004.

[UPP 01] UPPULURI P., SEKAR R., « Experiences with Specification-Based Intrusion Detection », LEE W., MÉ L., WESPI A., Eds., *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID)*, LNCS 2212, Springer-Verlag, 2001.

[VAL 01] VALDES A., SKINNER K., « Probabilistic Alert Correlation », LEE W., MÉ L., WESPI A., Eds., *Proceedings of RAID 2001, 4th International Symposium on Recent Advances in Intrusion Detection*, LNCS 2212, Springer Verlag, October 2001.

[VIG 96] VIGNA G., « A Topological Characterization of TCP/IP Security », rapport n° TR-96.156, 1996, Politecnico di Milano.

[VIG 99] VIGNA G., KEMMERER R. A., « NetSTAT : A Network-based Intrusion Detection System », *Journal of Computer Security*, vol. 7, n° 1, 1999.

[WOO 02] WOOD M., ERLINGER M., « Intrusion Detection Message Exchange Requirements », Internet draft (work in progress), February 2002, http ://search.ietf.org/internet-drafts/draft-ietf-idwg-requirements-06.txt.

Article reçu le 3 décembre 2002
Version révisée le 17 novembre 2003
Rédacteur responsable : Michael Rusinowitch

Fabien Autrel *est ingénieur ENSMA et effectue actuellement sa thèse au centre ONERA de Toulouse. Il travaille sur le regroupement, la fusion et la corrélation d'alertes de détection d'intrusions.*

Salem Benferhat *est professeur des universités à l'université d'Artois. Il effectue ses activités de recherche au CRIL (Centre de Recherche en Informatique de Recherche de Lens). Ses travaux de recherche se situent dans le domaine de l'intelligence artificielle et portent sur le raisonnement plausible et la dynamique des croyances en utilisant des modèles qualitatifs. Il fait partie du comité de programme de plusieurs conférences internationales (UAI, KR, AAAI, ISPTA, etc.).*

Frédéric Cuppens *est professeur à l'ENST Bretagne Campus de Rennes. Il s'intéresse à la modélisation en logique du premier ordre et en logique modale de problèmes relatifs à la sécurité des systèmes d'informations. Il est l'auteur de plus de 60 articles dans des revues et des conférences internationales à comité de lecture. Il a été organisateur ou président du comité de programme de plusieurs conférences internationales dans le domaine de la modélisation d'organisation et de la sécurité informatique.*

Hervé Debar*, expert sénior à France Télécom R&D, est spécialiste en techniques et outils de détection d'intrusions, ainsi que dans les technologies complémentaires (analyse de vulnérabilités, leurres, audits de sécurité, gestion des alertes et des incidents).*

Mireille Ducassé *est ingénieur de l'ENSEEIHT et docteur de l'Université de Rennes 1. Elle a passé dix ans dans des centres de recherche industriels et est professeur des universités à l'INSA de Rennes depuis 1993. Sa recherche est centrée sur le débogage automatisé et, par extension, sur l'analyse de traces et journaux. Elle est membre du comité de pilotage du séminaire international sur le débogage automatisé.*

Ludovic Mé *est ingénieur Supélec et docteur de l'Université de Rennes 1. Enseignant-chercheur à Supélec depuis 1988, il est depuis 1997 responsable de l'équipe de recherche SSIR (Sécurité des Systèmes d'Information et Réseaux). Il est l'auteur d'une vingtaine de communications dans le domaine de la détection d'intrusions et est membre du comité de pilotage du symposium RAID.*

Benjamin Morin *est ingénieur INSA et doctorant à France Télécom R&D. Sa thèse, soutenue début 2004, porte sur la corrélation d'alertes dans le domaine de la détection d'intrusions.*

Bernard Vivinis *est ingénieur Supélec. Enseignant-chercheur à Supélec Rennes depuis 1975, il est depuis 1997 membre de l'équipe de recherche SSIR (Sécurité des Systèmes d'Information et Réseaux).*

Rodolphe Ortalo *est ingénieur Supélec et docteur de l'INPT. Ses travaux de recherche ont porté sur l'évaluation quantitative de la sécurité des systèmes d'information et la formalisation des politiques de sécurité, puis sur la détection d'intrusion. Il est à présent responsable de la sécurité du système d'information de la CRAM Midi-Pyrénées.*