Agrégation d'alarmes faiblement structurées

Alexandre Vautier*, Marie-Odile Cordier*,
Mireille Ducassé**, René Quiniou***

*Irisa/Université de Rennes 1, **Irisa/Insa, ***Irisa/Inria
Campus de Beaulieu 35042 Rennes Cedex, France
{Alexandre.Vautier,Marie-Odile.Cordier, Mireille.Ducasse,Rene.Quiniou}@irisa.fr

Résumé. La contribution principale de ce document est une approche plaçant l'opérateur au coeur de l'analyse de journaux d'alarmes faiblement structurées en lui permettant d'utiliser ce qu'il sait, même si ses connaissances sont partielles, et sans le submerger d'informations. Des motifs temporels structurés sont extraits par agrégation d'alarmes généralisées et corrélation se basant sur la date des alarmes et sur la similarité d'attributs autres que la date. L'approche est appliquée aux alarmes produites par un concentrateur VPN (Virtual Private Network). Une étude de cas montre comment 5000 d'alarmes peuvent être regroupées en 50 motifs.

1 Introduction

La profusion des alarmes produites par les systèmes informatiques rend l'analyse des journaux d'alarmes de plus en plus difficile pour l'opérateur. Ces journaux sont générés, par exemple, par des composants réseau, des dispositifs de détection d'intrusion ou le système d'exploitation. Ils sont complexes et difficiles à appréhender complètement par un opérateur qui n'en a, en général, qu'une connaissance partielle. Confronté à cette masse de données, l'opérateur est, la plupart du temps, contraint de ne s'intéresser qu'à certaines alarmes qu'il sélectionne de façon plus ou moins pertinente. C'est, malgré tout, l'opérateur qui doit comprendre les informations contenues dans les journaux et qui sait de quoi il a besoin. Peu d'outils permettent de le seconder dans cette tâche risquée. L'augmentation grandissante des échanges d'informations, d'une part, et de la demande en sécurité, d'autre part, réclame la conception et la mise en œuvre de tels outils.

Les alarmes ainsi produites sont souvent faiblement structurées. Elles sont composées de champs mais les valeurs de ces champs sont souvent simplement copiées dans un fichier sous forme de chaînes de caractères sans marqueur syntaxique, reflétant le fait que l'analyse automatique de ces alarmes n'a pas été prévue lors de leur génération. Cet état de fait évolue, au moins pour les applications critiques, en particulier grâce à l'utilisation de XML. Cela étant, comme la structuration des données d'audit demande un certain effort, il reste encore de nombreuses applications où un journal est simplement un texte faiblement structuré. L'opérateur peut souvent paramétrer, au moins partiellement, les systèmes de génération des alarmes mais il en a rarement la maîtrise totale. De plus, c'est souvent à la suite d'un incident qu'il se penche sur un journal. Dans ce cas, il est trop tard pour paramétrer le système et il lui faut faire avec ce qui existe.

La contribution principale de ce document est de proposer une approche permettant de placer l'opérateur au coeur de l'analyse de ces journaux en lui permettant d'utiliser ce qu'il sait, même si ses connaissances sont partielles, et sans le submerger d'informations. Des motifs temporels sont

extraits de journaux faiblement structurés. Ces motifs sont des agrégations d'alarmes généralisées et corrélées. Dans une première étape, certains attributs d'alarmes sont extraits en s'appuyant sur des expressions régulières spécifiées par l'opérateur. Comme l'opérateur n'a pas forcément la connaissance pour spécifier directement les bonnes expressions, la spécification peut être affinée en plusieurs itérations. Dans une deuxième étape, les alarmes structurées issues de l'étape précédente sont regroupées. Les alarmes sont corrélées temporellement, en se basant sur la date des alarmes, et rationnellement, en se basant sur la similarité entre attributs autres que la date. On obtient ainsi une partition du journal.

Notre approche est appliquée aux alarmes produites par un concentrateur VPN (Virtual Private Network) de marque Cisco dans le réseau de France Télécom. Un concentrateur est un matériel réseau acceptant un grand nombre de connexions VPN simultanées. Dans ce cadre, les motifs temporels extraits constituent des modèles de connexion VPN présents implicitement dans le journal. Une connexion est composée de transactions, elles-mêmes constituées d'alarmes. Une étude de cas montre comment 5000 d'alarmes sont regroupées en 50 motifs.

L'extraction des attributs d'alarmes faiblement structurées est présentée dans la section 2. Le regroupement des alarmes est décrit dans la section 3. Les résultats d'une étude de cas sont présentés en section 4. Des travaux voisins sont discutés en section 5.

2 Extraction assistée des attributs

Les attributs sont générés à la suite de plusieurs interactions entre l'opérateur et un processus d'extraction automatique. À partir des résultats d'une extraction automatique et de ses propres connaissances, l'opérateur spécifie approximativement ou précisément la manière dont les attributs doivent être automatiquement extraits et le processus est réitéré. Les attributs générés sont utilisés dans la phase suivante (cf. section 3) pour construire des transactions.

Date	VPN	Type	Client	Groupe
Message				
05/09/2004 23 :11 :02.750	VPN-1	L2TP/46	82.83.84.85	
Tunnel to peer 82.83.84.85 closed, reason : Peer no longer responding				
05/09/2004 23 :12 :44.530	VPN-1	IKE/24	80.13.14.15	Group [Group1]
Received local Proxy Host data in ID Payload : Address 85.75.65.55, Protocol 17, Port 0				
05/10/2004 07 :46 :46.950	VPN-2	AUTH/37	20.21.22.23	
User [Unknown] Protocol [SNMP] attempted ADMIN logon.. Status : <ACCESS GRANTED> !				
06/02/2004 22 :37 :04.510	VPN-1	IKE/41		
IKE Initiator : Rekeying Phase 2, Intf 2, IKE Peer 81.71.61.51 local Proxy Address 85.75.65.55 , remote Proxy Address 81.71.61.51, SA (WindowsServer1)				
06/06/2004 15 :13 :45.640	VPN-1	IKE/41	81.251.55.54	Group [Group1]
IKE Initiator : Rekeying Phase 1, Intf 2, IKE Peer 81.251.55.54 local Proxy Address N/A, remote Proxy Address N/A, SA (N/A)				

FIG. 1 – *Exemples d'alarmes produites par le concentrateur VPN.*

Le journal étudié contient 1.262.117 alarmes issues du concentrateur VPN pendant environ 1 mois. Elles ont le format suivant : *date, nom de concentrateur VPN, type d'alarme, message* ainsi que les champs optionnels *adresse IP du client* et *groupe du client*. L'extraction des attributs est aisée pour tous les champs, sauf pour champ *message* qui, bien que le plus riche, n'est pas structuré :

```
- (\b(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}
                  (?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\b)(?:$|\W)
- (?:^|\W|0x)([a-fA-F0-9]{3,})(?:$|\W)
- User\s\[(\w*)\]
```

FIG. 2 – *Un ensemble $\mathcal{A}$ de trois expressions régulières : adresse IP constituée de 4 groupes de 2 à 3 chiffres, nombre hexadécimal supérieur à 99 et identificateur d'utilisateur.*

l'information contenue ne dépend pas seulement du type de l'alarme associée mais aussi de l'état du concentrateur. Ils contiennent des attributs de différents types (adresse IP, nombres hexadécimaux ou décimaux, nom d'utilisateur, etc.). La figure 1 présente quelques exemples d'alarmes du journal. Par exemple, la première indique qu'un incident de type L2TP/46 a eu lieu le 9 mai 2004 à 23h11min2.75s indiquant que le tunnel du client 82.83.84.85[1] a été fermé car il ne répondait plus.

Dans un premier temps, l'opérateur établit un ensemble $\mathcal{A}$ d'expressions régulières qui représentent la forme générale des attributs à extraire. La figure 2 montre trois expressions régulières[2] permettant de rechercher des adresses IP, des nombres comportant trois chiffres ou plus et des noms d'utilisateur. Chaque alarme est convertie en une *alarme normalisée* composée d'une date, d'un type et d'une liste d'attributs. La date et le type sont extraits des champs date et type de l'alarme. Les attributs des champs adresse IP du client, groupe du client et message sont extraits en utilisant $\mathcal{A}$. De cette façon, l'opérateur exprime approximativement ses connaissances dans $\mathcal{A}$, qu'il va pouvoir ensuite affiner au vu des résultats de l'extraction automatique. Les attributs et un ensemble $\mathcal{S}$ de *signatures d'alarme* sont ainsi générés.

Date	Type	Attributs
05/09/2004 23 :11 :02.750	L2TP/46	82.83.84.85
05/09/2004 23 :12 :44.530	IKE/24	80.13.14.15, 85.75.65.55
05/10/2004 07 :46 :46.950	AUTH/37	20.21.22.23
06/02/2004 22 :37 :04.510	IKE/41	81.71.61.51,85.75.65.55
06/06/2004 15 :13 :45.640	IKE/41	81.251.55.54

FIG. 3 – *Alarmes de la figure 1 converties en alarmes normalisées*

Définition 1 (Signature d'alarme)
Une signature est un triplet (t, l, e) où t est un type d'alarme, l une liste de types d'attribut et e une liste de doublets (r, f) formé d'une expression régulière r et d'une fonction $f : l \rightarrow \{0, 1\}$. Les expressions régulières r sont appliquées sur la concaténation des champs adresse IP du client, groupe du client et message pour extraire des alarmes de type t les attributs dont le type appartient à l. La première expression régulière r de la liste e aboutissant à une extraction est utilisée. $\forall x \in l$ si $f(x) = 1$ alors l'attribut correspondant au type x est présent dans l'expression r, sinon $f(x) = 0$ et cet attribut est absent de l'expression r.

Une signature d'alarme synthétise la façon dont les attributs ont été automatiquement extraits pour un type d'alarme donné. Elle peut être utilisée pour extraire les attributs de ce même type d'alarme lors de l'extraction automatique. L'opérateur examine ensuite les signatures générées et évalue si l'ensemble $\mathcal{A}$ est suffisant. Si tel n'est pas le cas pour un type d'alarme donné, il peut soit modifier l'ensemble $\mathcal{A}$, en ajoutant ou modifiant des expressions régulières, soit modifier

[1] Les noms d'utilisateur et les adresses IP apparaissant dans les exemples sont fictifs.
[2] Le format des expressions régulières est celui du package java.util.regex qui utilise la même syntaxe que perl.

la signature de ce type d'alarme dans $\mathcal{S}$. L'extraction automatique suivante est basée soit sur la signature extraite pour S si l'opérateur l'a décidé, soit sur l'ensemble $\mathcal{A}$ des formes d'attribut.

La figure 3 montre les alarmes normalisées construites à partir des alarmes de la figure 1 et de l'ensemble $\mathcal{A}$ de la figure 2. Notons que l'attribut 82.83.84.85 est présent dans le champ client et le champ message de l'alarme de type L2TP/46 mais un tel attribut n'est représenté qu'une fois dans l'alarme normalisée correspondante. Dans la suite du document, le terme *alarme normalisée* est substitué par le terme *alarme* quand il n'y a pas d'ambiguïté.

L'exemple suivant montre comment l'opérateur peut interagir avec le processus d'extraction. Les deux alarmes du type «IKE/41» de la figure 1 sont normalisées en deux alarmes de même type (voir figure 3) dont les listes de types d'attribut sont différentes. Les alarmes d'un même type doivent avoir la même liste de types d'attribut. L'examen, par l'opérateur, de la signature (figure 4) générée par l'extraction automatique lui montre que les alarmes de type IKE/41 ont en fait quatre attributs (et non un ou trois). Il peut alors modifier cette signature comme le montre la figure 5.

Plusieurs interactions entre l'opérateur et l'extraction automatique peuvent s'avérer nécessaires pour générer des alarmes normalisées utilisables dans l'étape de construction de transactions. Ainsi l'opérateur introduit, de façon aisée et incrémentale, ses connaissances sur le journal et améliore l'abstraction du journal en vue de la phase de construction des transactions.

Expression régulière *(IP) remplace une expression régulière*	Attributs IP	IP	IP
_ IKE Initiator : Rekeying Phase 2, Intf 2, IKE Peer (IP) local Proxy Address (IP), remote Proxy Address (IP), SA \(WindowsServer1\)	1	1	1
(IP) Group \[Group1\] IKE Initiator : Rekeying Phase 1, Intf 2, IKE Peer (IP) local Proxy Address N/A, remote Proxy Address N/A, SA \(N/A\)	1	0	0

FIG. 4 – *Signature générée pour les alarmes de type IKE/41. La colonne Attributs décrit la liste l , ici trois attributs de type IP. Les lignes de la matrice décrivent les éléments (r, f) de la liste e.*

Expression régulière *(IP) remplace une expression régulière*	Attributs IP	IP	IP	IP
_ IKE Initiator : Rekeying Phase 2, Intf 2, IKE Peer (IP) local Proxy Address (IP), remote Proxy Address (IP), SA \(WindowsServer1\)	0	1	1	1
(IP) Group \[Group1\] IKE Initiator : Rekeying Phase 1, Intf 2, IKE Peer (IP) local Proxy Address N/A, remote Proxy Address N/A, SA \(N/A\)	1	0	0	0

FIG. 5 – *Signature modifiée par l'opérateur pour les alarmes de type IKE/41*

3 Construction des transactions

Une connexion VPN entre un client et un concentrateur se compose de transactions réseaux qui provoquent des rafales d'alarmes apparaissant de manière isolée et dispersée dans le journal. Le but est de reconstruire les transactions et les connexions. Les alarmes d'une transaction sont proches dans le temps et partagent des attributs, par exemple, ceux propres au client ayant initié la transaction. Or, a priori, l'opérateur ne connaît pas ces attributs. C'est pourquoi, dans un premier temps *tous* les attributs sont considérés afin d'obtenir des transactions primitives. Celles-ci vont ensuite servir de base à la construction des transactions, assistée par l'opérateur.L'objectif de cette

Date	Type	Attributs
05/13/2004 14 :15 :50.910	IKE/25	82.81.80.79*
05/13/2004 14 :15 :50.910	IKE/24	82.81.80.79*, 85.75.65.55
05/13/2004 14 :15 :50.910	IKE/66	82.81.80.79*
05/13/2004 14 :15 :50.910	IKE/75	82.81.80.79*, 3600
05/13/2004 14 :15 :50.960	IKE/49	82.81.80.79*, 53e9fac2
05/13/2004 14 :15 :50.960	IKE/120	82.81.80.79*, 5da6fd05
05/13/2004 14 :15 :53.960	IKE/170	82.81.80.79*, d81626c
05/13/2004 14 :16 :18.140	IKEDBG/64	217.128.126.9*
05/13/2004 14 :16 :18.450	IKE/172	217.128.126.9*
05/13/2004 14 :16 :18.460	AUTH/12	483$^{\diamond}$
05/13/2004 14 :16 :18.560	AUTH/41	217.128.126.9*, 483$^{\diamond}$
05/13/2004 14 :16 :18.560	AUTH/13	483$^{\diamond}$
05/13/2004 14 :16 :18.630	IKE/79	217.128.126.9*, 6cf24368000000000f41
05/13/2004 14 :16 :23.320	AUTH/12	484^{+}
05/13/2004 14 :16 :23.430	AUTH/4	217.128.126.9*, 10.169.25.80, 484^{+}, user2

FIG. 6 – *Deux transactions primitives partitionnant une partie du journal*

construction est de réduire les informations présentes dans la synthèse présentées à l'opérateur sans pour autant réduire la quantité d'information. Ceci répond à l'objectif du Minimum Description Length (complexité de Kolmogorov, Vitanyi (2005)) i.e. la synthèse de taille minimale (vis à vis de l'opérateur) et perdant le moins d'information vis à vis du journal qu'elle résume.

3.1 Partition du journal en transactions primitives

Les transactions primitives sont extraites des alarmes normalisées du journal.

Définition 2 (α-transaction)
Une α-transaction est :
- soit, une séquence composée d'une alarme normalisée,
- soit, une séquence $< T, x >$ où T est une α-transaction et x une alarme telles qu'il existe une alarme y de T qui possède au moins n_a attributs en commun avec l'alarme x et l'intervalle de temps qui les sépare est inférieur à $maxGap$ (l'écart temporel maximum).

Définition 3 (Transaction primitive)
Une transaction primitive est une α-transaction maximale (aucune α-transaction la contient).

Les paramètres n_a et $maxGap$ sont fixés par l'opérateur. La figure 6 montre un exemple de découpage du journal en deux transactions primitives. Les attributs partagés qui permettent la construction de la transaction primitive sont identifiés par une même marque en exposant ($\diamond$,* ,$^{+}$).
.

Théorème 1
Pour tout journal d'alarmes normalisées et des paramètres n_a et $maxGap$ donnés, il existe un unique partitionnement définissant les transactions primitives représentant ce journal.

En faisant varier les paramètres n_a et $maxGap$, on peut générer un partitionnement comprenant autant de transactions primitives que d'alarmes normalisées jusqu'à un partitionnement

constitué d'une seule transaction primitive couvrant tout le journal. Il faut donc souvent contraindre plus fortement le partitionnement afin de satisfaire le critère de MDL. C'est le rôle de la phase suivante qui permet de construire des modèles de transactions à partir des transactions primitives.

3.2 Construction des modèles de transactions

Nous définissons tout d'abord un modèle d'alarme afin de pouvoir définir un modèle de transaction qui correspond à une généralisation, soit des alarmes, soit des transactions, par extension de leur domaine au niveau des attributs. Une relation d'ordre partiel sur de tels modèles est ensuite introduite. Enfin, la génération des modèles de transaction est décrite.

Définition 4 (Modèle d'alarme)
Un modèle d'alarme est un couple (t, l) où t est un type d'alarme et l est une liste de variables. Une variable $v \in l$ possède un type t_v et son domaine d_v est contraint.

Il faut noter que le temps n'est pas représenté explicitement dans un modèle d'alarme.Le temps est considéré comme un élément permettant d'ordonner les alarmes au niveau d'un modèle de transaction. Un modèle d'alarme couvre (généralise) des alarmes appelées instances de ce modèle d'alarme. Par exemple, l'alarme (date = 05/13/2004 14 :19 :46.940, type = PPPDECODE/16, attributs = [80.14.52.129, user1, 85.75.65.55]) est une instance du modèle d'alarme (t = PPPDECODE/16, $l = [X, Y, Z]$) où X et Z sont des variables de type adresse IP qui correspondent respectivement aux adresses 80.14.52.129 et 85.75.65.55 et Y est une variable de type identifiant qui correspond à l'attribut user1. L'ensemble des valeurs que peuvent prendre les variables est restreint par leur domaine associé. Par exemple, au modèle précédent on peut ajouter la contrainte $X \in$ 80.14.*.* qui impose à la variable X de prendre ses valeurs entre $80.14.0.0$ et $80.14.255.255$.

Définition 5 (Modèle de transaction)
Un modèle de transaction est un doublet (S,V) où S est une séquence de modèles d'alarme et V est l'ensemble des variables apparaissant dans les modèles d'alarme de la transaction.

Type	Variables
IKEDBG/64	X
IKE/172	X
AUTH/12	Z
AUTH/41	X, Z
AUTH/13	Z,
IKE/79	X, W
AUTH/12	A
AUTH/4	X, B, A, C
$X[IP] \in 217.128.*.*, B[IP], W[int], Z[int], A[int], C[String]$	

FIG. 7 – *Ce modèle de transaction généralise la seconde transaction de la figure 6. Les lignes du tableau sont les éléments de S et la dernière ligne décrit les variables de V et leur domaine.*

Un modèle de transaction couvre (généralise) des transactions appelées instances.Ainsi, la deuxième transaction de la figure 6 est une instance du modèle de transaction de la figure 7. Ce dernier comporte une variable X partagée par 5 modèles d'alarme. Une telle variable apparaissant dans plusieurs modèles d'alarme de la transaction contraint les attributs correspondant des instances à posséder un type (un domaine) identique.

Type	Variables
IKEDBG/64	X
AUTH/12	Z
AUTH/41	X, Z
AUTH/13	Z
$X[IP] \in 217.*.*.*, Z[int]$	

FIG. 8 – *Un modèle de transaction généralisant le modèle de transaction de la figure 7*

De plus, une relation de généralité peut être définie sur les modèles de transaction. Ainsi, le modèle de transaction de la figure 8 est plus général que le modèle de la figure 7.

Définition 6 (Relation d'ordre partiel sur les modèles d'alarme)
Soient A_g et A_s deux modèles d'alarme. A_g est plus général que A_s si et seulement si le type de A_g est identique au type de A_s et si pour toute variable v_g de A_g il existe une variable v_s de A_s telles que le domaine de v_s est inclus dans le domaine de v_g.

Définition 7 (Relation d'ordre partiel sur les modèles de transactions)
Un modèle de transaction $T_g = (S_g, V_g)$ est plus général qu'un modèle de transaction $T_s = (S_s, V_s)$ si et seulement si S_g est une sous-séquence de S_s telle que chaque alarme de S_g est plus générale que l'alarme correspondante de S_s.

Les modèles de transaction sont générés à partir du partitionnement décrit dans la partie 3. Les transactions primitives contenant des alarmes de type identique et des attributs éventuellement différents mais de même type sont généralisées en un modèle unique : le modèle le plus spécifique généralisant ces transactions (aussi appelé *lgg* - least general generalisation Plotkin (1970)). Ainsi, les types des modèles d'alarme de ce modèle sont identiques à ceux des transactions primitives et les domaines des variables du modèle sont aussi contraints que possible.

Par exemple, le modèle de la figure 8 est le lgg des deux transactions de la figure 9. Le domaine de la variable X du lgg est contraint à être le plus petit possible, soit $217.*.*.*$.

$t1$			$t2$		
Date	Type	Attributs	Date	Type	Attributs
...	IKEDBG/64	217.128.16.9	...	IKEDBG/64	217.56.200.1
...	AUTH/12	483	...	AUTH/12	513
...	AUTH/41	217.128.16.9, 483	...	AUTH/41	217.56.200.1, 513
...	AUTH/13	483	...	AUTH/13	513

FIG. 9 – *Le modèle de transaction de la figure 8 est le lgg des transactions primitives t_1 et t_2.*

4 Étude de cas et perspectives

Les 1.262.117 alarmes du journal du concentrateur VPN ont été entièrement normalisées après trois passages de l'extraction automatique des attributs. 37 signatures générées automatiquement ont dû être modifiées et 4 formes d'attribut ont été ajoutées aux expressions régulières de la figure 2. Les modifications de signatures font apparaître qu'un dispositif d'inférence grammaticale pourrait être utile. En effet, en corrélant les expressions régulières d'une même signature, il serait

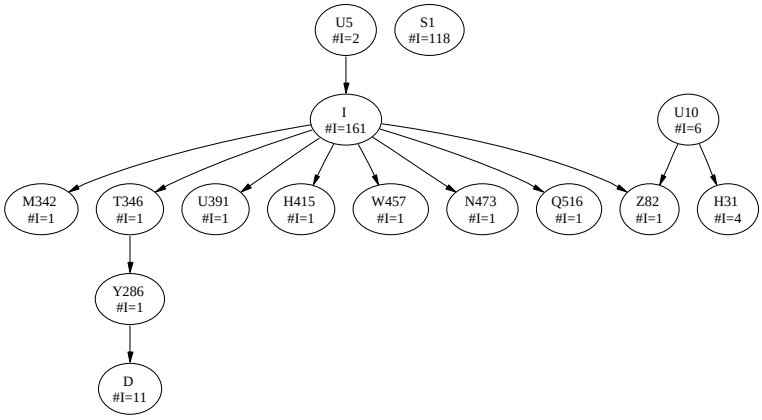

FIG. 10 – *Une partie du graphe des modèles de transaction extraits à partir des 5000 premières alarmes du journal (*$maxGap = 10s, n_a = 1$*)*

possible d'associer automatiquement les attributs correspondants L'exemple d'interaction entre l'opérateur et la base de signatures présenté en partie 2 pourrait ainsi être automatisé.

La construction des transactions, expérimentée sur les 5000 premières alarmes du journal avec $maxGap = 10s$ et $n_a = 1$, a produit 628 transactions primitives qui ont généré 81 modèles de transaction. La figure 10 montre une partie du graphe généré à partir de cette construction : un nœud représente un modèle de transaction et affiche le nombre d'instances qui lui sont associées, un arc représente la relation de généralité. Si le modèle A est plus général que le modèle B alors il existe un arc de A vers B. On constate que le modèle de transaction nommé «I» (dont une instance est la première transaction de la figure 6) est très fréquent, de même que le modèle «S1» constitué de 8 modèles d'alarme. Ces deux modèles couvrent 41% de la partie de journal analysé.

80% des modèles n'ont qu'une seule instance et devraient être fractionnés en modèles plus petits ou fusionnés avec d'autres modèles. Certains modèles de transaction ont été générés par des variables qui n'étaient pas propres à un client mais plutôt à des serveurs, d'autres par des identifiants communs à toutes les alarmes, etc. Une analyse statistique sur les variables des modèles de transactions pourrait déterminer celles qui participent à la construction de transactions primitives mais qui ne doivent pas participer à la construction des transactions finales. Un modèle de transaction généré à partir de l'une de ces variables doit être fractionné. Enfin, certaines transactions primitives ne présentant aucun attribut commun ont des occurrences proches temporellement dans le journal pourraient être fusionnées.

5 Travaux voisins

La profusion des alarmes provenant des outils de détection d'intrusions ou d'attaques nécessite un pré-traitement pour fournir à l'opérateur de sécurité une vision synthétique des alarmes. De nombreux travaux présentent des méthodes visant à enrichir, structurer des informations provenant d'alarmes ou corréler les alarmes ainsi obtenues. La méthode présentée ici étant proche de la fouille de données, nous nous focalisons sur des travaux ayant des liens avec ce thème.

La plupart des travaux n'abordent pas l'étape de préparation des données et considèrent que l'ensemble des attributs utilisés pour décrire les alarmes et les domaines associés sont connus *a priori*. Par exemple, les données d'alarmes fournies par les IDS sont parfois au format XML[3] ce qui permet une analyse syntaxique simple et l'intégration des données dans une base de données relationnelle, par exemple. Nous proposons une méthode d'assistance à l'opérateur pour constituer un ensemble d'attributs à partir de données faiblement structurées.

Une première étape pour le résumé ou la synthèse des alarmes consiste à les regrouper ou les agréger. L'agrégation locale vise à regrouper, sur une fenêtre temporelle de taille limitée, les alarmes issues de la même attaque. L'agrégation globale s'attache à caractériser un type d'attaque en regroupant, sur la totalité de la base, les alarmes ayant des attributs communs. Un groupe peut ensuite être *synthétisé* en méta-alarme puis *qualifié* en adjoignant des informations contextuelles, telles que le type de réseau ou les équipements observés, contribuant à améliorer leur sémantique. Différents critères basés sur une notion de similarité sont utilisés pour l'agrégation : similarité probabiliste (pondérée) basée sur la proximité des valeurs d'une partie (Dain et Cunningham (2001)) ou de la totalité des attributs (Valdes et Skinner (2001)), règles expertes définissant une similarité logique (Cuppens (2001)). Julisch (2003) agrège les alarmes contenues dans une table relationnelle selon leur type. Les attributs des alarmes d'un type donné sont généralisés selon des taxonomies fournies par des opérateurs tant que le nombre d'occurrences de l'alarme en cours de généralisation n'atteint pas un certain seuil. Morin (2004) systématise cette approche en traitant la corrélation d'alarmes comme un processus de recherche d'information : un treillis de concepts (Ganter et al. (2005)) représentant les alarmes contenues dans le journal est construit de manière incrémentale. Un concept de ce treillis est une méta-alarme synthétisant un sous-ensemble d'alarmes.

Dans notre approche, une alarme est associée à une transaction si elle possède un certain nombre d'attributs pouvant être mis en relation avec des attributs des alarmes de la transaction dans une fenêtre temporelle limitée. L'égalité des valeurs est généralement utilisée mais une notion de similarité étendue, par exemple basée sur une taxonomie telle que celle employée par Julisch, pourrait être introduite, ce qui nous rapprocherait de la méthode de Cuppens (2001). L'explicitation de relations logiques entre les attributs permettent, à notre avis, d'expliquer plus clairement les liens entre les alarmes à l'opérateur que des statistiques sur l'occurrence des attributs des alarmes.

L'étape d'agrégation est généralement suivie d'une étape de corrélation qui met en évidence des épisodes ou chroniques, encore appelés plans d'intrusion, constitués d'une ou plusieurs séquence d'alarmes élémentaires ou de méta-alarmes. Klemettinen et al. (1999) utilise en guise de corrélation d'alarmes la méthode proposée par Mannila et al. (1997) en fouille de données temporelles pour extraire des règles d'association et des épisodes sur la base de motifs temporels fréquents. Qin et Lee (2003) utilisent le test de causalité de Granger (1969) pour regrouper des méta-alarmes sans connaissances *a priori* des propriétés des alarmes. Cette approche paraît séduisante mais nécessite des données volumineuses contenant de nombreuses méta-alertes pour être utilisable. Dans notre approche, à toutes les transactions sont associées des statistiques sur leur occurrence mais c'est l'opérateur qui décide si une transaction doit être retenue ou non. Cuppens et Miège (2002) utilisent une notion de corrélation logique. Une alarme A est corrélée à l'alarme B si les conséquences de A rendent l'alarme B exécutable ce qui signifie qu'un ou plusieurs attributs des deux alarmes sont similaires. Cuppens et Miège prévoient une génération automatique des règles de corrélation d'alarmes mais le processus n'est pas détaillé. Nous utilisons également une notion de corrélation *relationnelle* mais basée sur une relation de causalité s'appuyant uniquement sur la séquentialité temporelle, plus simple à mettre en œuvre de manière semi-automatique.

[3] http ://xml.coverpages.org/idmef.html

6 Conclusion

Ce document décrit une préparation assistée d'un opérateur humain d'un journal d'alarmes issues d'un concentrateur VPN. La principale difficulté est le manque de structure explicite et d'informations sur les attributs des alarmes. L'opérateur interagit avec les différents modules afin d'élaborer une synthèse du journal et d'acquérir de nouvelles connaissances sur celui-ci.

L'analyse effectuée n'est pas propre aux journaux de concentrateur VPN mais est généralisable à d'autres journaux faiblement structurés : syslog, trace de programme, etc. Les techniques utilisées, l'extraction des attributs à partir de leur forme, l'agrégation en transactions primitives composées d'alarmes corrélées relationnellement et la construction des modèles de transactions, sont toutes généralisables à d'autres problèmes.

Références

Cuppens, F. (2001). Managing alerts in a multi-intrusion detection environment. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC)*.

Cuppens, F. et A. Miège (2002). Alert correlation in a cooperative intrusion detection framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*.

Dain, O. et R. Cunningham (2001). Fusing a heterogeneous alert stream into scenarios. In *Proceedings of the 2001 ACM Workshop on Data Mining for Security Applications*.

Ganter, B., G. Stumme, et R. Wille (2005). *Formal Concept Analysis, Foundations and Applications*. Springer Verlag.

Granger, C. (1969). Investigating causal relations by econometric methods and cross-spectral methods. *Econometrica 34*, 424–438.

Julisch, K. (2003). Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security 6*(4).

Klemettinen, M., H. Mannila, et H. Toivonen (1999). Rule discovery in telecommunication alarm data. *Journal of Network and Systems Management 7*(4).

Mannila, H., H. Toivonen, et A. I. Verkamo (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery 1*(3), 259–289.

Morin, B. (2004). *Corrélation d'alertes issues d'outils de détection d'intrusions avec prise en compte d'informations sur le système surveillé*. Ph. D. thesis, INSA de Rennes.

Plotkin, G. (1970). A note on inductive generalization. In *Machine Intelligence*, Volume 5, pp. 153–163. Edinburgh University Press.

Qin, X. et W. Lee (2003). Statistical causality analysis of infosec alert data. In *RAID 2003*, Volume 2820 of *LNCS*, pp. 73–93. Springer Verlag.

Valdes, A. et K. Skinner (2001). Probabilistic alert correlation. In *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, Volume 2212 of *LNCS*. Springer Verlag.

Vitanyi, P. M. B. (2005). *Algorithmic statistics and Kolmogorov's Structure Functions*, pp. 151–174. MIT Press.

Summary