Nouvelle Approche de Corrélation d'Alertes basée sur la Fouille Multidimensionnelle

Hanen BRAHMI, Imen BRAHMI, Sadok BEN YAHIA

Faculté des Sciences de Tunis.
Département des Sciences de l'Informatique.
Campus Universitaire 1060.
{hanenbrahmi; imen.brahmi}@gmail.com, sadok.benyahia@fst.rnu.tn

Résumé. En réponse aux problèmes posés par la complexité croissante des réseaux et des attaques, les Systèmes de Détection d'Intrusions (SDIs) constituent une bonne alternative pour mieux sécuriser un système informatique. Cependant, les SDIs existants présentent des lacunes en terme de génération excessive d'alertes. Réellement, la majorité de ces alertes ne correspondent pas à des attaques (fausses alertes, alertes redondantes, etc.). Ainsi, la corrélation d'alertes est un processus d'analyse appliqué à des journaux d'alertes. Dans cet article, nous proposons une nouvelle approche pour la corrélation d'alertes basée sur le couplage entre la fouille de données et les outils OLAP (*On Line Analytical Processing*). L'idée intuitive derrière cette approche est de profiter des avantages de la fouille de données multidimensionnelles afin de rehausser l'analyse des alertes et introduire une solution puissante pour faire face aux défauts des SDIs. Les expérimentations, que nous avons menées, montrent l'efficacité de notre nouvelle méthode de corrélation d'alertes.

1 Introduction

Avec le développement accru des réseaux de communication, les risques causés par les attaques sur les systèmes informatiques deviennent un réel problème pour les entreprises et les organisations. Par conséquent, afin de protéger les systèmes d'éventuelles attaques, de nouvelles approches appelées *Systèmes de Détection d'Intrusions* (SDI) ont fait leur apparition. Ces outils ont pour objectif d'analyser le traffic réseau et de détecter les comportements malveillants (Singhal et Jajodia, 2010).

Le revers de la médaille de l'utilisation des SDIs réside dans la génération excessive des alertes. La majorité de ces dernières ne correspondent pas réellement à des attaques (fausses alertes, alertes redondantes, etc.). En effet, le volume des données contenues dans un journal d'alertes est très important. De plus, il augmente d'une manière très rapide puisque plusieurs giga-octets d'alertes peuvent s'accumuler par jour sur certains systèmes (Sadoddin et Ghorbani, 2006). Afin de faire face à ce problème, des approches de *corrélation d'alertes* ont été proposées.

La *corrélation d'alertes* est l'analyse des alertes déclenchées par un ou plusieurs SDIs afin de fournir une vue synthétique et de haut niveau des événements malveillants intéressants

ciblant le système d'information (Sadoddin et Ghorbani, 2006). Une alerte est un message textuel généré par un SDI quand une attaque ou une activité anormale est détectée. Il contient le temps de la détection, la signature de l'alerte, le protocole, l'adresse IP de l'attaquant, l'adresse IP du victime, etc. Supposons qu'un opérateur de sécurité se charge d'analyser un nombre impressionnant d'alertes afin de prendre des décisions appropriées. Il est clair qu'il va être noyé dans cette masse de connaissances et se retrouvera rapidement débordé.

Parallèlement, la gestion des grandes masses de données est devenue une tâche difficile et assez coûteuse à maintenir. Ce problème a conduit au développement de *la fouille de données multidimensionnelles* (OLAP Mining), qui correspond à un processus de fouille de données intégrant une composante OLAP (*On Line Analytical Processing*). Les techniques de corrélation d'alertes basées sur l'entrepôt de données et l'OLAP auront de nombreux avantages. Premièrement, ces derniers fourniront une représentation de haut niveau des alertes avec un respect de la relation temporelle dans la production de ces alertes. Deuxièmement, ils offriront un moyen efficace pour distinguer une vraie alerte d'une fausse alerte. Troisièmement, ils peuvent être utilisés pour prévoir les prochaines étapes d'une attaque et, par conséquent, arriver à une stratégie visant à réduire les dommages (Singhal et Jajodia, 2010).

Dans cet article, nous examinons une autre façon afin d'attaquer le problème de corrélation d'alertes. Ainsi, nous introduisons une nouvelle approche de corrélation fondée sur une perspective d'entrepôt de données visant l'amélioration de l'analyse des journaux d'alertes. Cette nouvelle proposition du couplage entre l'analyse en ligne et la fouille de données se base sur une approche qui adapte un algorithme d'extraction de règles d'association au contexte multidimensionnel. En effet, nous modélisons les données relatives aux alertes sous forme d'une structure multidimensionnelle, appelée le *cube d'alertes*. Ensuite, nous introduisons un nouvel algorithme qui fournit une représentation concise de règles d'association multidimensionnelles extraites directement à partir du cube d'alertes sans transformation préalable de ce dernier. De plus, notre algorithme proposé permet d'offrir à l'utilisateur la possibilité de guider le processus de fouille vers des contextes d'analyse ciblés qui répondent à ses besoins d'explication et à partir desquels seront extraites les règles d'association. Les expérimentations menées sur un journal d'alertes réelles prouvent l'efficacité de notre proposition.

Le reste de l'article est organisé comme suit. Nous introduisons notre approche dans la section 2. Les résultats des expérimentations montrant l'utilité de l'approche proposée sont présentés dans la section 3. La conclusion et les travaux futurs font l'instance de la section 4.

2 Corrélation d'alertes basée sur la fouille multidimensionnelle

La corrélation d'alertes basée sur la fouille multidimensionnelles vise à aboutir à un SDI fiable avec une amélioration de détection et une génération réduite de fausses alertes. En première étape, un cube d'alertes est construit en utilisant les dimensions disponibles. Avec les techniques développées pour la construction des cubes de données, les administrateurs de sécurité, qui utilisent OLAP, sont largement capables d'explorer les alertes, de naviguer dans les niveaux hiérarchiques des dimensions et d'en extraire des informations intéressantes selon plusieurs niveaux de granularité. Cependant, la technologie OLAP se limite à des tâches exploratoires et ne fournit pas d'outils automatiques pour expliquer les relations et les associations

potentiellement existantes entre les données d'un cube. Ainsi, en deuxième étape, nous couplons la technologie OLAP avec l'extraction des règles d'association pour en tirer des règles de détection plus efficaces.

Dans la suite, nous nous concentrons sur l'étude de ces deux étapes.

2.1 Cube d'alertes : Modélisation, construction et manipulation

Nous proposons de modéliser les données des journaux d'alertes comme une structure multidimensionnelle fondée sur le schéma en étoile illustrée dans la Figure 1.

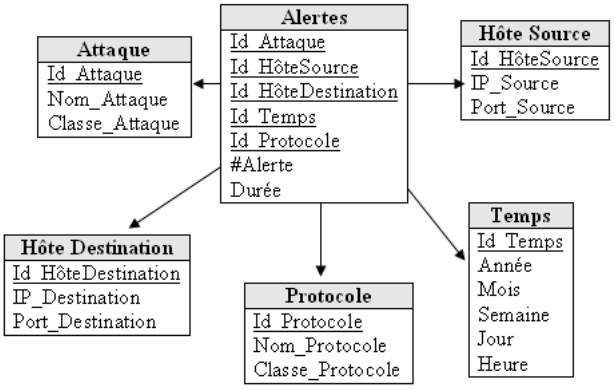

FIG. 1 – *Schéma en étoile d'un cube d'alertes.*

La table de faits "Alertes" contient l'attribut "#Alerte" qui mesure le nombre d'alertes. La dimension "Temps" comprend des informations concernant la date et l'heure de capture du paquet sur le réseau. La dimension "Protocole" contient le nom et la classe de protocole qui a été attaqué. "Hôte Source" décrit l'adresse IP et le numéro du port de la source. De même, la dimension "Hôte Destination" décrit l'adresse IP et le numéro du port de la destination. Enfin, la dimension "Attaque" contient le nom de l'attaque et son type.

Afin de profiter de la maniabilité de la structuration multidimensionnelle, les analystes de sécurité doivent s'attaquer à la prise en compte des hiérarchies. La prise en compte des informations à différents niveaux d'abstraction permet de rehausser les capacités d'analyse. En effet, les alertes sont souvent décrites par des relations hiérarchiques. Pour ce faire, nous définissons une hiérarchie de concepts pour chaque dimension dans le cube d'alertes. Par exemple, "Heure $\rightarrow$ Jour $\rightarrow$ Semaine $\rightarrow$ Mois $\rightarrow$ Année" est la hiérarchie sur la dimension "Temps". La dimension "Attaque" peut être organisées dans la hiérarchie "Nom $\rightarrow$ Classe". En outre, les hiérarchies peuvent être pré-définies ou générées par le partitionnement de la dimension en intervalles. Ainsi, la dimension "Durée" pourrait être partitionnée en plusieurs catégories comme "Faible", "Moyenne" et "Élevée".

En utilisant le schéma en étoile décrit dans la Figure 1, nous obtenons un cube d'alertes à six dimensions.

Une cellule du cube obtenu contient les agrégats des opérations sur les mesures. Par exemple, une cellule pourrait correspondre à une courte durée d'attaques sur le service FTP pour le 20 Octobre 2011. Le cube d'alertes peut être construit en utilisant les fonctions d'agrégation du

langage SQL comme (COUNT, SUM, MIN, MAX, etc.). Par exemple, la valeur de la fonction COUNT se réfère au nombre d'alertes. En outre, l'OLAP complète les entrepôts de données en proposant des outils pour l'analyse, la visualisation et la navigation dans les cubes de données. Ainsi, la manipulation des alertes peut se faire avec une grande flexibilité. En effet, ces dernières peuvent être visualisées à partir de perspectives différentes par l'utilisation de cubes de données. Les opérateurs OLAP, comme (ROLL-UP, DRILL-DOWN, SLICE et DICE), offrent des capacités analytiques qui peuvent être appliquées sur les données des journaux d'alertes.

Bien que la technologie OLAP améliore l'analyse, elle se limite à des tâches exploratoires et ne fournit pas d'outils automatiques pour expliquer les relations et les associations potentiellement existantes entre les données d'un cube. Par exemple, un utilisateur peut noter, à partir d'un cube d'alertes, que le niveau d'alerte de l'attaque "*SNMP missing*" est particulièrement élevé sur une adresse IP_Destination donnée. En revanche, cette exploration ne permet pas d'expliquer automatiquement les raisons de ce fait particulier. En effet, pour arriver à expliquer l'ordre de certains faits OLAP ou des phénomènes particuliers, un utilisateur est habituellement amené à explorer manuellement et observer l'ensemble des données selon plusieurs axes d'analyse. Par exemple, le niveau élevé des alertes des attaques "*SNMP missing*" peut être expliqué par son association à un protocole et à une adresse IP_Source égale à "172.16.0.1".

2.2 Extraction de règles d'association multidimensionnelles

L'extraction de règles d'association est l'un des principaux problèmes de la fouille de données. Il a été introduit par (Agrawal et al., 1993) dans le but d'analyser les bases de données de transactions de ventes. Il a pour but de découvrir des relations significatives entre les données de la base de données. Étant donné une base de données de transactions de ventes, constituée d'une liste d'articles achetée par un client, une règle associative est une relation d'implication $X \Rightarrow Y$ entre deux ensemble d'articles X et Y, qui satisfait des seuils minimums du support et de confiance spécifiés par l'utilisateur, *c.-à-d.*, $minSup$ [1] et $minConf$ [2]. X est appelé *prémisse*, alors que Y est *la conclusion* de la règle (Agrawal et al., 1993).

Dans les dernières années, beaucoup d'études ont abordé le problème de l'extraction des règles d'association à partir des cubes de données. Chaudhuri et Dayal (1997) ont mis en exergue l'importance de l'exploration des cubes de données en employant les algorithmes d'extraction des règles d'association. Ross et Srivastava (1997) avancent que la recherche des règles d'association peut interagir avec les outils OLAP afin d'extraire automatiquement des connaissances à partir des cubes de données. En effet, les agrégats de comptage (COUNT) nécessaires pour la recherche des règles d'association sont déjà pré-calculés dans un cube de données. De plus, les hiérarchies des dimensions du cube peuvent être exploitées afin d'extraire des règles à plusieurs niveaux de granularité. Imieliński et al. (2002) défendent le même point de vue et considèrent que l'OLAP est étroitement lié avec les règles d'association. Ils pensent également que l'extraction des connaissances est un objectif commun à la technologie OLAP et la recherche des règles d'association. En conclusion, il est claire que l'extraction des règles d'association peut rendre OLAP plus utile et plus facile à appliquer dans le schéma global des systèmes décisionnels (Ben Messaoud et al., 2008).

1. $minSup$ est le seuil minimal du support pré-défini par l'utilisateur.
2. $minConf$ est le seuil minimal de confiance pré-défini par l'utilisateur.

Les règles d'association multidimensionnelles s'avèrent être utiles pour augmenter la précision de détection et diminuer le taux des faux positifs (Brahmi et al., 2012; P.-Ping et Q.-Ping, 2002). Par conséquent, la performance d'un SDIs peut être considérablement améliorée si les règles d'association sont extraites à partir d'un cube d'alertes. Néanmoins, le nombre des règles extraites peut être assez important, ce qui affecte la vitesse du SDI et nuit sa performance (P.-Ping et Q.-Ping, 2002). Certaines de ces règles sont redondantes puisqu'elles contiennent des motifs qui correspondent à des sous-ensembles d'autres motifs.

Exemple 1 *Soit R et R_1 deux règles d'association multidimensionnelles. R : {Port_Source = 6677 $\wedge$ IP_Destination = 192.63.11.11 $\wedge$ Protocole = TCP $\wedge$ Duration = Long} $\Rightarrow$ {Attaque = WEB-CGI} et R_1 : {Port_Source = 6677 $\wedge$ Protocole = TCP} $\Rightarrow$ {Attaque = WEB-CGI}. R et R_1 partagent des caractéristiques similaires, c.-à-d., les motifs "Port_Source = 6677" et "Protocole = TCP". Si les supports de ces deux motifs sont égaux, alors la règle R_1 est redondante par rapport à R.*

Afin d'extraire efficacement les règles d'association multidimensionnelles non-redondantes à partir du cube d'alertes, nous utilisons le concept de fermeture (Pasquier et al., 1999) défini comme suit :

Définition 1 *Soit γ l'opérateur de fermeture affectant à un motif X son sur-ensemble maximal ayant la même valeur du support que X. Un motif X est fermé s'il n'existe pas de motif X' tel que : (i) X' est un sur-ensemble propre de X ; et (ii) tous les enregistrements de connexion dans un trafic réseau contenant X contiennent également X'.*

À cet égard, nous introduisons l'algorithme AMAR (*Audit Multidimensional Association Rules mining*) destiné à extraire une représentation concise de règles d'association multidimensionnelles à partir d'un cube d'alertes $\mathcal{CA}$. Le pseudo-code qui résume notre démarche générale est illustré par l'algorithme 1.

Habituellement l'utilisateur est intéressé par des sous-ensembles précis d'attributs afin d'en extraire des relations intéressantes entre eux. Ainsi, il/elle a besoin d'exclure l'ensemble des attributs non pertinents de l'examen. À cette fin, notre algorithme AMAR permet à l'utilisateur de guider le processus d'analyse par : (i) la définition de l'ensemble des dimensions $\mathcal{D}$ à analyser ; (ii) le choix des niveaux des hiérarchies $\mathcal{H}_{\mathcal{D}}$ associés aux dimensions d'analyse, et (iii) le réglage des seuils $minSup$ et $minConf$.

Comme il est illustré par l'algorithme 1, nous adoptons une stratégie itérative ascendante pour la recherche des k-motifs fermés fréquents, où k est un indice correspondant à l'itération en cours de l'algorithme (nombre d'items dans un motif). Nous désignons par C_k l'ensemble de k-motifs candidats, CF_k l'ensemble de k-motifs fermés candidats qui peuvent être fréquents et CFF_k l'ensemble de k-motifs fermés fréquents.

L'extraction des motifs fermés fréquents est effectuée niveau par niveau. Ainsi, durant **une étape d'initialisation** (ligne 2), notre algorithme capture les 1-candidats C(1) à partir des dimensions d'analyse $\mathcal{D}$ définies par l'utilisateur dans le cube d'alertes $\mathcal{CA}$. Les éléments de C(1) correspondent aux attributs de $\mathcal{D}$, où chacun se conforme avec les hiérarchies choisies $\mathcal{H}_{\mathcal{D}}$. Dans la **première étape**, AMAR applique le concept de fermeture (*cf.* Définition 1). La **deuxième étape** (lignes 9-12) de notre algorithme dérive les motifs fermés fréquents CFF_k à partir des motifs fermés candidats CF_k qui ont un support supérieur ou égal à $minSup$. La **troisième étape** consiste à extraire les règles d'association avec une confiance supérieure ou

égale à $minConf$. Les calculs du support et de la confiance sont effectués respectivement par les fonctions CALCULSUPPORT et CALCULCONFIANCE. Les deux fonctions cherchent directement les agrégats nécessaires précalculés à partir du cube de données via les requêtes MDX (MultiDimensional eXpression) (Ben Messaoud et al., 2008). La **quatrième étape** (lignes

Algorithme 1 : L'algorithme AMAR.

Données : $\mathcal{CA}$, $\mathcal{D}$, $\mathcal{H}_{\mathcal{D}}$, $minSup$, $minConf$.
Résultat : Ensemble de règles d'association multidimensionnelles non-redondantes, *c.-à-d.*, $\mathcal{X} \Rightarrow \mathcal{Y}$, avec les valeurs Supp et Conf correspondantes.

```
1  début
2    C_1 := {1-candidat} ;
3    k := 1 ; /* |1-candidat| est la cardinalité des attributs correspondant à D et H_D.*/
4    tant que C_k ≠ ∅ et k ≤ |1-candidat| faire
5      CF_k := ∅ ;
6      CFF_k := ∅ ;
7      pour chaque motif candidat A ∈ C_k faire
8        CF_k := CF_k ∪ γ(A);
9      pour chaque motif candidat fermé A ∈ CF_k faire
10       Supp := CALCULSUPPORT(A);
11       si Supp ≥ minSup alors
12         CFF_k := CFF_k ∪ A;
13     pour chaque A ∈ CFF_k faire
14       pour chaque B ≠ ∅ et B ⊂ A faire
15         Conf := CALCULCONFIANCE(A − B, B);
16         si Conf ≥ minConf alors
17           X := A − B;
18           Y := B;
19           retourner (X ⇒ Y, Supp, Conf);
20     C_{k+1} := ∅ ;
21     /* CFF_k.générateur est l'ensemble des générateurs des motifs fermés fréquents de taille
       k.*/
22     pour chaque A ∈ CFF_k.générateurs faire
23       pour chaque B ∈ CFF_k.générateurs qui partage (k-1) items avec A faire
24         si Tout Z ⊂ {A ∪ B} ayant k items est un motif inter-dimensionnel et est fermé
           fréquent alors
25           C_{k+1} := C_{k+1} ∪ {A ∪ B} ;
26     k := k + 1;
27 fin
```

22-25) utilise l'ensemble des k-motifs générateurs des fermés fréquents CFF$_k$.générateurs pour obtenir un nouvel ensemble de $(k+1)$-candidats, noté C$_{k+1}$. Un $(k+1)$-candidat est l'union de deux k-motifs $\mathcal{A}$ et $\mathcal{B}$ dans CFF$_k$.générateurs qui respectent trois conditions : (i) $\mathcal{A}$ et $\mathcal{B}$ doivent partager $k-1$ motifs communs ; (ii) tous les sous-motifs non vides de $\mathcal{A} \cup \mathcal{B}$ doivent

être des instances de motifs inter-dimensionnels [3] de $\mathcal{D}$; et (iii) tous les sous-motifs non vide dérivés à partir de $\mathcal{A} \cup \mathcal{B}$ doivent être des motifs fréquents.

TAB. 1 – *Un aperçu d'un cube d'alertes à quatre dimensions sous forme tabulaire.*

Protocole	Src _Port	Dst _Port	Attaque	#Alertes
TCP	63587	80	*WEB-MISC*	44
TCP	6161	80	*WEB-CGI*	26
UDP	6161	110	*WEB-MISC*	15
UDP	63587	80	*WEB-CGI*	20
ICMP	63587	1	*WEB-MISC*	64

TAB. 2 – *Liste des règles d'association multi-dimensionnelles.*

ID	Règles	$\mathcal{S}$up	$\mathcal{C}$onf
R_1	80⇒ *WEB-CGI*	0.3	0.5
R_2	80∧ 63587 ⇒ TCP ∧ *WEB-MISC*	0.3	0.7
R_3	*WEB-CGI* ⇒ TCP ∧ 80∧ 6161	0.2	0.6
R_4	63587 ∧ *WEB-MISC* ⇒ ICMP ∧ 1	0.4	0.6
R_5	UDP ∧ 80⇒ 63587 ∧ *WEB-CGI*	0.1	1.0
R_6	UDP ∧ 63587 ⇒ 80∧ *WEB-CGI*	0.1	1.0

Exemple 2 *Soit le Tableau 1 présentant un exemple d'un cube d'alertes à quatre dimensions. La dernière ligne mesure le nombre d'alertes à l'aide de la fonction d'agrégation* COUNT. *L'ensemble des motifs fermés fréquents, avec leurs supports correspondants, est le suivant :* {(*"UDP" : 0.2*), (*"80" :0.5*), (*"63587" :0.7*), (*"6161" : 0.2*), (*"WEB-MISC" : 0.7*), (*"UDP, 80, 63587, WEB-CGI" : 0.1*), (*"UDP, 110, 6161, WEB-MISC" : 0.08*), (*"63587, 80" : 0.3*), (*"80, WEB-CGI" : 0.2*), (*"TCP, 80" : 0.4*), (*"63587, WEB-MISC" : 0.6*), (*"TCP, 80, 6161, WEB-CGI" : 0.1*), (*"TCP, 80, 63587, WEB-MISC" : 0.2*), (*"ICMP, 1, 63587, WEB-MISC" : 0.3*)}. *Nous extrayons l'ensemble des règles d'association multidimensionnelles en utilisant l'algorithme* AMAR. *Tout au long de notre exemple, nous avons fixé* $minSup$ *à 10% et* $minConf$ *à 50%. L'algorithme a généré 40 règles. Certaines des règles extraites sont illustrées dans le Tableau 2.*

3 Évaluation expérimentale

Pour évaluer l'efficacité de notre méthode de corrélation d'alertes basée sur la fouille multi-dimensionnelle, nous avons mené une série d'expérimentations sur un PC équipé d'un Pentium 4 avec une fréquence d'horloge de 3 GHz et une mémoire RAM de 2 Go, utilisant la distribution de Linux Fedora Core 6 comme système d'exploitation.

Durant les expérimentations effectuées, nous avons utilisé un log d'alertes réelles générées par le SDI SNORT [4] issues de "*Inter-Service Academy Cyber Defense Competition*" [5] et capturées durant la période de Novembre 2008 à Novembre 2011. Pour construire le cube de données d'audit, nous adoptons le schéma en étoile montré dans la Figure 1. La construction du cube d'alertes est effectuée en utilisant l'outil "Analysis Services of SQL Server 2008". Notre étude expérimentale comprend deux volets :

- Premièrement, nous comparons le temps d'extraction des règles d'association mutidimensionnelles non-redondantes consommé par AMAR contre celui de l'algorithme OLEMAR [6] introduit par (Ben Messaoud et al., 2008).

- Deuxièmement, nous nous concentrons sur l'évaluation de la fiabilité du SDI fondé sur la détection d'attaques en utilisant les règles multidimensionnelles générées par AMAR.

3. Un motif inter-dimensionnel est composé d'attributs provenant de différentes dimensions.
4. SNORT est un SDI gratuit de source ouverte disponible à http ://www.snort.org/.
5. http ://www.itoc.usma.edu/research/dataset/
6. Nous tenons à remercier Riadh Ben Messaoud pour nous avoir fourni le code source de l'algorithme OLEMAR.

3.1 Performance d'AMAR

La figure 2(A) montre le temps d'exécution de notre algorithme AMAR par rapport à OLEMAR, en fonction de la variation du support minimum $minSup$. En général, nous remarquons que le temps d'exécution des deux algorithmes décroît en fonction du $minSup$. Plus $minSup$ est élevé, plus les deux algorithmes deviennent rapides.

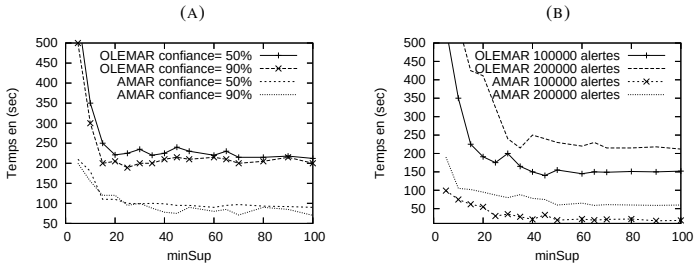

FIG. 2 – *Temps d'exécution des algorithmes OLEMAR et AMAR.*

En revanche, il est claire que le temps d'exécution d'AMAR est très inférieur à celui d'OLEMAR. Ces résultats s'expliquent par le fait que le nombre des motifs fréquents est important et la proportion des motifs fermés fréquents parmi ces derniers est faible. L'espace de recherche de l'algorithme AMAR qui adapte l'algorithme CLOSE (Pasquier et al., 1999) est de taille très inférieure à celle d'OLEMAR qui adapte l'algorithme APRIORI (Agrawal et al., 1993). De plus, nous notons que, lorsque $minConf$ augmente, le temps d'exécution des deux algorithmes baisse globalement. Cependant, contrairement au $minSup$, $minConf$ n'influence pas d'une manière sensible la rapidité des algorithmes.

La figure 2(B) résume les tests de performance des algorithmes AMAR et OLEMAR pour des cubes d'alertes de différents volumes en fonction du $minSup$. Chaque cube est caractérisé par le nombre des alertes qu'il contient. En effet, nous remarquons que quelque soit le volume du cube étudié AMAR reste toujours plus rapide qu'OLEMAR. Pour les valeurs faibles du support minimal (moins de 40%), le nombre des alertes étudiées est un élément déterminant dans la rapidité des deux algorithmes. En revanche, pour les valeurs élevées du $minSup$, le nombre des alertes n'a pratiquement aucune influence sur le temps d'exécution des algorithmes. En effet, quelque soit le nombre d'alertes, les algorithmes gardent globalement le même temps d'exécution.

3.2 Fiabilité du SDI

La prise en compte de la fiabilité des SDIs est une question cruciale pour des tâches de corrélation d'alertes. À cet égard, afin de prouver l'importance et l'efficacité des règles générées par l'algorithme AMAR, nous nous sommes intéressés à mesurer le *Taux de Détection* (TD) [7] d'un SDI basé sur ces règles.

Le Tableau 3 fournit des détails sur les attaques du journal d'alertes utilisées dans nos expérimentations. Parmi les attaques détectées par SNORT, nous avons sélectionné sept attaques Web. Toutes ces attaques ciblent soit des serveurs Web soit des applications Web associées.

7. Le *Taux de Détection* (TD) mesure combien d'attaques sont détectées correctement.

TAB. 3 – *Distributions des attaques dans les jeux d'apprentissage et de test.*

		Jeu d'apprentissage	Jeu de test
Signature	**Nom SNORT de l'alerte**	**#**	**#**
1091	Attaque1 : WEB-MISC ICQ Webfront HTTP DOS	87	6
2002	Attaque2 : WEB-PHP remote include path	50	231
2229	Attaque3 : WEB-PHP viewtopic.php access	5169	1580
1012	Attaque4 :WEB-IIS fpcount attempt	3	10
1256	Attaque5 : WEB-IIS CodeRed v2 root.exe access	2	3
1497	Attaque6 : WEB-MISC cross site scripting attempt	5602	7347
2436	Attaque7 : WEB-CLIENT Microsoft wmf metafile access	145	53

Le TD du SDI basé sur les règles générées par AMAR (symbolisé par SDI-OLAP) en fonction de la variation des dimensions, pour les sept différentes attaques, est montré par le Tableau 4. La variation des dimensions a été aboutie en utilisant l'algorithme AMAR.

TAB. 4 – *Le TD (%) du SDI basé sur les règles générées par AMAR à l'égard de la variation des dimensions.*

Dimensions	**Attaque1**	**Attaque2**	**Attaque3**	**Attaque4**	**Attaque5**	**Attaque6**	**Attaque7**
2-D	96.8	86.4	66.6	74.9	90.3	81.6	63.2
3-D	97.9	83.2	67.8	76.7	91.9	82.8	72.3
4-D	98.2	91.1	69.8	79.5	93.2	88.6	79.5
5-D	98.5	**95.3**	71.5	81.3	**95.6**	91.2	**84.4**
6-D	**99.5**	95.2	**74.9**	**86.6**	93.9	**93.8**	82.3

D'après les résultats, nous pouvons remarquer que le cube d'alertes à six dimensions donne les meilleures performances pour détecter les attaques. Comme exemple, le cube d'alertes détecte l'attaque1 avec TD 99,5% alors que le cube d'alertes à cinq dimensions donne le pire TD avec 98,5%. Par conséquent, il est évident que SDI-OLAP permet la détection des attaques avec le meilleur TD autant que le nombre de dimensions est le plus élevé, *c.-à-d.*, six dimensions. Même si, le TD diminue en fonction de la diminution du nombre de dimension, il reste élevé.

4 Conclusion et perspectives

L'analyse en ligne, la corrélation d'alertes et la fouille de données sont trois champs de recherche qui ont connu depuis quelques années, des évolutions parallèles. Le travail mené dans ce papier montre l'importance et l'intérêt de l'association entre ces trois domaines scientifiques. Dans le but d'améliorer la corrélation d'alertes générées par un SDI, nous avons conçu une nouvelle méthode qui se base sur une structure de données multidimensionnelle appelée cube d'alertes. Ensuite, nous avons introduit l'algorithme AMAR permettant d'extraire l'ensemble des règles d'association non-redondantes à partir de ce cube. Les expérimentations menées montrent l'efficacité d'AMAR ainsi que du SDI basé sur les règles générées par cet algorithme. Comme un futur travail, nous proposons d'améliorer l'efficacité de l'extraction en s'appuyant sur d'autres représentations condensées des connaissances extraites (non-dérivables, fermés non-dérivables).

Remerciements

Ce travail a été partiellement financé par le projet CMCU 11G1417.

Références

Agrawal, R., T. Imielinski, et A. Swami (1993). Mining Association Rules Between Sets of Items in Large Databases. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data (SIGMOD 1993), Washington, USA*, pp. 207–216.

Ben Messaoud, R., S. L. Rabaséda, R. Missaoui, et O. Boussaid (2008). OLEMAR *:* A*n* O*nline* E*nvironment for* M*ining* A*ssociation* R*ules in* M*ultidimensional* D*ata*, Volume 2, pp. 14–47.

Brahmi, H., I. Brahmi, et S. Ben Yahia (2012). OMC-IDS : At the Cross-Roads of OLAP Mining and Intrusion Detection. In *Proceedings of the 16th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Kuala Lumpur, Malaysia*. To appear.

Chaudhuri, S. et U. Dayal (1997). An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record 26*(1), 65–74.

Imieliński, T., L. Khachiyan, et A. Abdulghani (2002). Cubegrades : Generalizing Association Rules. *Data Mining and Knowledge Discovery 6*(3), 219–258.

P.-Ping, M. et Z. Q.-Ping (2002). Association Rules Applied to Intrusion Detection. *Wuhan University Journal of Natural Sciences 7*(4), 426–430.

Pasquier, N., Y. Bastide, R. Taouil, et L. Lakhal (1999). Efficient Mining of Association Rules Using Closed Itemset Lattices. *Journal of Information Systems 24*(1), 25–46.

Ross, K. et D. Srivastava (1997). Fast Computation of Sparse Data Cubes. In *Proceedings of the 23rd International Conference on Very Large Databases (VLDB'97), Athens, Greece*, pp. 116–125.

Sadoddin, R. et A. Ghorbani (2006). Alert Correlation Survey : Framework and Techniques. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust : Bridge the Gap Between PST Technologies and Business ServicesMarkham, Ontario, Canada*, pp. 37 :1–37 :10.

Singhal, A. et S. Jajodia (2010). Data Mining for Intrusion Detection. In O. Maimon et L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook*, pp. 1171–1180. Springer.

Summary

Due to the growing threat of network attacks, the efficient detection as well as the network abuse assessment are of paramount importance. In this respect, the Intrusion Detection Systems (IDS) are intended to protect information systems against intrusions and attacks. However, IDS are plugged with several problems that slow down their development, such as the excessive generation of alerts. The majority of these alerts do not really correspond to attacks (false alarms, redundant alerts, etc..). Therefore, alert correlation is a process of analysis applied to the logs of alerts in order to reduce their impressive number. In this paper, we introduce a novel approach of alert correlation, which integrates data mining techniques and On Line Analytical Processing (OLAP) tools. The main idea behind this approach is to take advantage of OLAP mining to enhance the alert analysis and introduce a powerful solution to deal with the defects of IDS. Our experiment results show the robustness and efficiency of our new method for alert correlation.