Réseaux Bayésiens naïfs pour la détection des attaques coordonnées

Salem Benferhat, Tayeb Kenaza, Aïcha Mokhtari

Réseaux Bayésiens naïfs pour la détection des attaques coordonnées

Salem Benferhat [*] — **Tayeb Kenaza** [*] — **Aïcha Mokhtari** [**]

[*] *CRIL (CNRS-UMR 8188), Université d'Artois*
Rue Jean Souvraz SP 18 F-62307 Lens Cedex
{benferhat,kenaza}@cril.univ-artois.fr

[**] *Département informatique, USTHB, BP 32, El Alia, 16000 Alger*
aissani_mokhtari@yahoo.fr

RÉSUMÉ. *La corrélation d'alertes est un mécanisme indispensable pour la réduction du volume important des alertes et pour la détection des attaques coordonnées et complexes. Les approches existantes soit se basent sur des connaissances d'experts, soit utilisent des simples mesures de similarité qui ne permettent pas de détecter des attaques complexes. Elles souffrent également d'une complexité de calcul très élevée dû par exemple à un grand nombre de scénarios possibles pour détecter une attaque coordonnée. Dans cet article, nous proposons une approche de corrélation d'alertes basée sur les réseaux bayésiens naïfs. Notre modélisation implique une légère contribution des connaissances d'experts. Elle tire profit des données disponibles, et fournit des algorithmes efficaces pour la détection et la prédiction des scénarios les plus plausibles. Notre approche est illustrée en utilisant les bases de données DARPA 2000.*

ABSTRACT. *Alert correlation is a very useful mechanism to reduce the high volume of reported alerts and to detect complex and coordinated attacks. Existing approaches either require a large amount of expert knowledge or use simple similarity measures that prevent detecting complex attacks. They also suffer from high computational issues due, for instance, to a high number of possible scenarios. In this paper, we propose a naive bayes approach to alert correlation. Our modeling only needs a small part of expert knowledge. It takes advantage of available historical data, and provides efficient algorithms for detecting and predicting most plausible scenarios. Our approach is illustrated using the well known DARPA 2000 data set.*

MOTS-CLÉS : *Détection d'intrusions, Corrélation d'alertes, Prédiction d'attaques, Réseaux Bayésiens naïfs*

KEYWORDS: *Intrusion detection, Alert correlation, Attack prediction, Naive bayes.*

1re soumission à *JFRB*, le 28 février 2008

1. Introduction

Les systèmes de détection d'intrusions (IDS) sont généralement considérés comme une seconde ligne de défense pour protéger contre les activités malicieuses. Les IDSs traditionnels se concentrent habituellement sur la détection des attaques élémentaires. Ils traitent les alertes séparément et indépendamment sans tenir compte des relations qui puissent exister entre elles. Le résultat des IDS est généralement un ensemble d'alertes qui rapportent des attaques élémentaires.

Toutefois, des intrus peuvent utiliser des attaques complexes pour atteindre leurs objectifs. Souvent, ils effectuent une série d'actions (attaques élémentaires) dans une séquence bien définie, appelée "scénario" ou "plan d'attaque". La plupart de ces actions sont signalées par les IDSs, mais les relations logiques entre ces actions (séquence d'actions) ne sont pas détectées par les outils standards. Ainsi, les administrateurs système sont souvent submergés par un volume important d'alertes à corréler manuellement. Á cette fin, l'objectif de la corrélation est de rechercher des relations entre les alertes.

La corrélation d'alertes a été étudiée ces dernières années par plusieurs chercheurs. On peut distinguer deux principales catégories d'approches :

1) La première catégorie se concentre sur la réduction du volume d'alertes soit en utilisant des mesures de similarité entre des attributs tels que : la classification des attaques, les adresses source et cible, l'identité des utilisateurs, le temps de détection, etc (Valdes *et al.*, 2001), soit en utilisant des mécanismes d'agrégation d'alertes (Cuppens, 2001)(Debar *et al.*, 2001) ou des mécanismes de clustering (Julisch, 2001).

2) La deuxième catégorie cherche à détecter les relations entre les actions pour découvrir des scénarios d'attaque soit en utilisant les préconditions et les postconditions des actions pour construire implicitement des scénarios d'attaque (Cuppens *et al.*, 2002)(Steven *et al.*, 2000)(Ning *et al.*, 2002), soit tout simplement en introduisant la description des scénarios dans le système (Dain *et al.*, 2001).

Les méthodes existantes permettent de réduire le volume d'alertes et de détecter les plans d'attaque achevés. Cependant, pendant la prédiction d'attaques ces méthodes génèrent un nombre important, voire exponentiel, de scénarios, ce qui rends très difficile aux administrateurs d'analyser chaque scénario. En plus, elles impliquent une grande contribution des connaissances d'experts soit pour donner des descriptions complètes des scénarios d'attaque, soit pour définir les préconditions et les postconditions des attaques élémentaires.

Dans cet article, nous présentons une nouvelle approche de corrélation d'alertes basée sur les Réseaux Bayésiens (RB) naïfs permettant la détection des attaques coordonnées. Les RB naïfs représentent une forme simple des réseaux bayésiens, qui sont des modèles graphiques permettant d'utiliser efficacement des informations incertaines. Les RB naïfs ont été utilisés dans des nombreuses applications, notamment dans la détection d'intrusions (Abouzakhar *et al.*, 2003)(Axelsson, 2004)(Gowadia *et al.*, 2005)(Krügel *et al.*, 2003)(Puttini *et al.*, 2003). Cependant, peu de travaux ont ap-

pliqué les RB à la corrélation d'alertes (Geib *et al.*, 2001)(Qin *et al.*, 2004). En fait, les quelques travaux existants qui appliquent les réseaux bayésiens pour détecter des attaques coordonnées nécessitent que des connaissances sur les scénarios (sous forme d'arbre de décision) soient préalablement définis. Dans notre approche une telle représentation explicite des scénarios n'est pas nécessaire, et nous n'avons même pas besoin de déterminer explicitement l'ensemble des actions impliquées dans un scénario. Tout est obtenue à partir des données d'observations.

Notre approche est efficace pour la prédiction des scénarios d'attaque et elle n'implique pas une grande contribution des connaissances d'experts. Le processus de corrélation d'alertes sera considéré dans cet article comme un problème de classification. Étant donné un ensemble d'actions récemment observées et un ensemble d'objectifs d'intrusion, notre but est de déterminer les objectifs d'intrusion les plus plausibles. Lorsque les observations ne favorisent aucun plan d'attaque, notre approche est également en mesure de confirmer que le trafic est normal.

Le reste de cet article est organisé comme suit : la section 2 introduit les RB naïfs. Dans la section 3, nous présentons le problème de la corrélation d'alertes. La section 4 présente notre approche en trois phases : le prétraitement des données d'observations, la construction des RB naïfs et le processus de prédiction des objectifs d'intrusion. Nous illustrons également notre approche sur les données DARPA 2000. La dernière section conclut l'article.

2. Rappel sur les Réseaux Bayésiens

Les réseaux bayésiens sont l'un des modèles graphiques largement utilisés pour représenter et manipuler des informations incertaines (Jensen, 1996)(Pearl, 1991). Ils sont constitués de deux composants :

– Un composant graphique représenté par un graphe acyclique orienté (DAG) dont les nœuds représentent les événements et les arcs représentent les relations entre ces événements.

– Un composant numérique qui consiste en une quantification des différents liens dans le graphe par une distribution des probabilités conditionnelles de chaque nœud dans le contexte de ses parents.

Les RB naïfs (Shachter *et al.*, 1992) représentent une forme très simple des réseaux bayésiens, qui se composent d'un graphe avec un seul parent, représentant le nœud non observé, et plusieurs nœuds feuilles correspondant au nœuds observés, avec une forte hypothèse d'indépendance entre les feuilles dans le contexte de leur parent.

Les RB naïfs ont donné des résultats satisfaisants des problèmes de classiification (Friedman *et al.*, 1996). La classification est assurée en considérant le nœud parent (racine) comme une variable non observée qui représente la classe d'un objet, et les nœuds feuilles comme étant des variables observées correspondants aux différents attributs spécifiant cet objet.

Par conséquent, en présence d'un ensemble d'apprentissage, la seule investigation à faire est de calculer les probabilités conditionnelles puisque la structure du réseau est unique. Ce calcul peut être résumé comme suit :

– les probabilités conditionnelles pour les attributs discrets sont principalement calculées à partir des fréquences en comptant le nombre d'apparitions de chaque valeur d'attribut avec chacune des valeurs que le nœud parent peut prendre ;

– les attributs continus sont généralement traités en supposant qu'ils suivent une distribution de probabilité gaussienne (c'est-à-dire normale). Donc, pour chaque valeur de classe c_i et chaque attribut continu A_k, nous devons calculer la moyenne μ et l'écart type σ qui vont nous servir pour le calcul de la fonction de densité de probabilité pour chaque valeur a_k de A_k comme suit :

$$f(a_k) = \frac{1}{\sqrt{2\pi}.\sigma}.e^{-\frac{(a_k-\mu)^2}{2\sigma^2}}$$

L'hypothèse de normalité peut être considérée comme une restriction des réseaux bayésiens naïfs puisque certains attributs peuvent ne pas suivre une distribution normale. Dans ce cas, nous pouvons utiliser une méthode non-paramétrique telle que l'estimation de densité par le noyau qui ne suppose aucune distribution particulière pour les attributs continus (Duda *et al.*, 2000). Cette méthode est basée sur la localisation pour chaque a_k d'un attribut continu A_k les observations qui lui sont voisines à travers une fonction de pondération $K_\sigma(a_k, a_i)$ qui affecte un poids à chaque instance $a_i \in D_{A_K}$ basé sur sa distance par rapport à a_k. Le choix le plus commun de $K\sigma$ est le noyau Gaussien $K\sigma = \phi|a_i - a_k|/\sigma$, où σ est l'écart type. Une autre alternative serait de simplement discrétiser les attributs continus.

Une fois le réseau bayésien est quantifié, il est possible de classer tout nouvel objet, étant donné les valeurs des attributs, en utilisant la règle de Bayes exprimées par :

$$P(c_i|A) = \frac{P(A|c_i).P(c_i)}{P(A)},$$

où c_i est une valeur possible de la classe et A représent l'observation concernant les attributs. Soit $a_1, a_2, ..., a_n$ les valeurs observées des attributs $A_1, A_2, ..., A_n$. Sous l'hypothèse que les attributs sont indépendants (dans le contexte du noeud parent C), la probabilité $P(c_i|A)$ peut être développée comme suit :

$$P(c_i|A) = \frac{P(a_1|c_i).P(a_1|c_i)...P(a_n|c_i).P(c_i)}{P(a_1, a_2, ..., a_n)}$$

Notons qu'il n'est pas nécessaire de calculer explicitement le dénominateur $P(a_1, a_2, ..., a_n)$, car il est déterminé par normalisation.

3. Les attaques coordonnées et la corrélation d'alertes

Durent la surveillance des systèmes d'informations (SI), les IDSs génèrent des alertes lorsque des actions suspectes sont observées. Certaines actions ne peuvent pas être observées pour différentes raisons telles que : des actions se trouvant hors champ d'observation ou des IDS qui ne sont pas fiables, etc. Les alertes rapportées chaque jour représentent des instanciations d'un ensemble fini d'actions modélisées dans le système. Par exemple, des centaines d'alertes "ICMP ping" peuvent être générées, après un scan du réseau, et qui représentent des instances d'une même action "scan". Comme nous le verrons plus loin dans notre approche, les actions vont représenter les variables d'intérêt de notre RB naïf.

Généralement, un intrus effectue des actions dans un ordre bien défini appelé "plan d'attaque". Dans un plan d'attaque, les premières actions modifient un SI ou fournissent des information à un intrus, en vue d'accomplir les dernières actions. Un plan d'attaque est modélisé comme un processus de planification d'actions qui transforment un SI d'un état à un autre, jusqu'à ce qu'il atteint un certain état cible, que nous appelons "Objectif d'intrusion" (Cuppens *et al.*, 2002). Pour déterminer cette séquence, certaines approches utilisent le mécanisme de précondition et postcondition, qui nécessite une grande contribution des connaissances d'experts afin de définir les préconditions et les postconditions des actions. De plus, dans (Cuppens *et al.*, 2002) lorsque certaines actions ne sont pas observées, certaines alertes virtuelles sont générées. Cela augmente le nombre de scénarios possibles, et la corrélation d'alertes pondérée proposée dans (Benferhat *et al.*, 2003) limite seulement les conséquences de cette explosion du nombre de scénarios.

Notre approche ne nécessite pas de déterminer au préalable les préconditions et les postconditions des actions. Elle permet de prédire directement les plus plausibles objectifs d'intrusion en utilisant l'historique des observations. En fait, nous ne sommes pas intéressés à déterminer l'ordre exact dans lequel un ensemble d'actions ont été exécutées de manière à atteindre un objectif d'intrusion. Nous sommes plus intéressés d'une part à déterminer quelles sont les actions peuvant être impliquées dans un objectif d'intrusion, et d'autre part de développer un outil qui permet de prédire, en temps réel, quel objectif d'intrusion peut être compromis. Il est très important de noter que notre approche ne nécessite pas des connaissances d'experts, plus préséciment elle n'exige ni les préconditions et les postconditions des actions comme dans (Cuppens *et al.*, 2002)(Ning *et al.*, 2002)(Steven *et al.*, 2000), ni une représentation explicite des scénarios d'attaque comme dans (Dain *et al.*, 2001). Elle ne nécessite même pas l'ensemble des actions impliquées dans les attaques. En fait, cet ensemble sera déterminé automatiquement en se basant sur les données d'apprentissage.

Dans la suite, nous utilisons une définition faible d'un plan d'attaque, qui est définie comme étant un ensemble $S = \{A_1,\ A_2,\ ...,\ A_n,\ O\}$, dont les A_i représentent des instances d'actions et O est un objectif d'intrusion tel que chaque A_i a une *influence* sur O. Une définition possible de l'influence est : A_i a une influence sur O si $P(O|A_i) > P(O)$.

Certaines actions peuvent être impliquées dans plusieurs plans d'attaque, et certains objectifs peuvent être atteints par plusieurs plans d'attaque. Par exemple, un Déni de service (DoS) peut être effectué par un simple Ping de la mort ou Synflood, ou par une attaque plus sophistiquée comme Smurf. L'objectif de notre approche est de détecter les plans d'attaque le plus tôt possible et de prévoir les plus plausibles. Étant donné un objectif d'intrusion, nous pouvons distinguer trois types d'actions :

– Actions avec influence négative qui diminuent la probabilité d'atteindre l'objectif d'intrusion, tel que : $P(O|A_i) < P(O)$

– Actions avec influence positive qui augmentent la probabilité de compromettre l'objectif d'intrusion sans vraiment y parvenir, tel que : $P(O|A_i) > P(O)\ et\ P(O|A_i) < Seuil$. Cela signifie que la probabilité d'atteindre l'objectif d'intrusion augmente sans dépasser certain seuil (50% par exemple).

– Actions avec influence critique qui permettent d'atteindre directement l'objectif d'intrusion, tel que : $P(O|A_i) > P(O)\ et\ P(O|A_i) > Seuil$. Cela signifie que la probabilité d'atteindre l'objectif d'intrusion dépasse un certain seuil.

La section suivante présente notre approche sur l'application des réseaux bayésiens naïfs pour détecter des attaques coordonnées.

4. Modélisation de la corrélation d'alertes par les réseaux bayésiens naïfs

Dans cette section, nous expliquons comment modéliser la corrélation d'alertes par les RB naïfs, en exploitant l'historiques des observations. Notre approche comprend trois étapes principales :

1) **Prétraitement des données d'observations :** cette étape concerne le prétraitement de l'historique des observations. Le résultat de cette étape est un ensemble de données formatées.

2) **Construction du RB naïf :** dans cette étape, nous estimons la distribution des probabilités de chaque variable du RB naïf.

3) **Prédiction des objectifs d'intrusion :** dans cette étape nous prédisons les objectifs d'intrusion par l'application des mécanismes d'inférence des RB.

Notre approche sera illustrée sur le premier scénario des données DARPA 2000 (DARPA-2000, 2000). Le premier scénario DARPA 2000 comprend un déni de service distribué (DDoS) mené par un attaquant novice. Le but de cette attaque est qu'un attaquant relativement novice cherche à démontrer ses performances à l'aide d'une attaque en "scripte" pour compromettre plusieurs hôtes sur Internet, installer les composants nécessaires pour mener un DDoS, et ensuite lancer un DDoS contre un site gouvernementale. Dans cette attaque l'adversaire exploite une faille dans l'outil Sadmind (outil d'administration à distance) pour obtenir un accès root dans trois hôtes Solaris du site Eyrie Air Force Base (AFB) (DARPA-2000, 2000). Les phases du scénario d'attaque sont :

1) scan du site AFB à partir d'un site distant

2) recherche des adresses IP des hôtes Solaris exécutant Sadmind

3) compromission des hôtes via la vulnérabilité du Sadmind

4) installation du trojan mstream DDoS sur les trois hôtes du site AFB

5) Lancement du DDoS

Dans la phase 1, l'intrus effectue un IPsweep de plusieurs sous-réseaux sur le site AFB. Il envoie des requêtes ICMP-echo dans ce balayage et écoute les répences ICMP-echo afin de déterminer quels hôtes sont en place. Dans la phase 2, les hôtes découverts sont interrogés pour déterminer ceux qui exécutent Sadmind. Lors de la phase 3, l'intrus tente de compromettre les hôtes exécutant Sadmind. L'attaquant tente d'exploiter Sadmind plusieurs fois dans chaque hôte, chaque fois avec des paramètres différents. Á la fin de cette phase, l'intrus obtient un accès root sur trois hôtes. Dans la phase 4, l'attaquant effectue une connexion telnet sur les hôtes compromis et installe les composants nécessaires pour le DDoS (mstrem serveur et mstream client). En dernière phase, l'intrus lance le DDoS contre la victime.

Nous allons maintenant décrire les trois étapes de notre approche.

4.1. *Prétraitement des données d'observations*

Pour construire le RB naïf, nous allons effectuer un certain prétraitements sur les données d'observations. Les données contiennent un ensemble d'alertes qui rapportent les actions exécutées et également des informations sur les objectifs d'intrusion (s'ils ont été atteints ou non). Nous allons d'abord regrouper les objectifs d'intrusion observés dans une seule classe appelée "Objectifs-Intrusion" et nous affectons à chaque objectif un numéro entre 0 à n, où 0 représente aucun objectif d'intrusion et N représente le nombre maximum d'objectifs à protéger. Ainsi, le domaine d'Objectifs-Intrusion est {0, 1, 2, ... N}. Par exemple, le premier scénario de DARPA 2000 contient un seul objectif, une attaque DDoS, la classe va contenir alors deux valeurs $\mathrm{Dom}(classe) = \{0, 1\}$ (0 signifie que l'objectif n'est pas atteint, et 1 signifie que l'objectif est compromis).

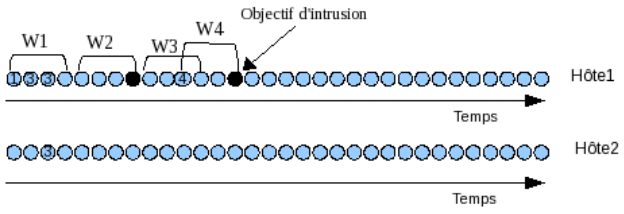

Figure 1. *Prétraitement des données d'observations*

L'étape suivante consiste à trier les alertes observées en fonction de leur ordre chronologique de détection. Nous les regroupons en sous groupes, en fonction d'une

certaine fenêtre de temps déterminée expérimentalement (de quelques minutes jusqu'à 02 heures). Ces fenêtres constituent habituellement le temps nécessaire pour achever un plan d'attaque (voir la figure 4.1). Ces fenêtres sont cruciales pour déterminer l'ensemble des actions impliquées dans les scénarios.

Si un objectif d'intrusion est observé dans une fenêtre, nous déplaçons cette fenêtre à gauche jusqu'à ce qu'elle se termine sur cet objectif (voir la figure 4.1. Nous faisons cela afin de s'assurer que toutes les actions impliquées dans chaque objectif d'intrusion sont présentes dans une même fenêtre. En procédant de cette façon signifie que certaines actions seront considérées sur deux fenêtres simultanément. Dans la figure 4.1 par exemple, l'action 4 appartient aux fenêtres W3 et W4. W3 contient un trafic normale et W4 contient un plan d'attaque (car à la fin de la fenêtre 4 un objectif d'intrusion est compromis). En fonction de la fréquence d'observation de l'action 4 sur des trafics normaux ou anormaux, nous pouvons déterminer si l'action 4 est suspecte ou non.

Enfin, nous étiquettons chaque sous groupe par le numéro correspondant à l'objectif d'intrusion observé. En cas où aucun objectif n'a été observé, le numéro 0 est utilisé pour dire que ce trafic ne contient pas de plan d'attaque. Il est possible d'observer plus d'un objectif sur une même fenêtre, dans ce cas certains sous groupes peuvent être étiquetés avec plusieurs numéros. Ainsi, nous obtiendrons les observations sous forme de vecteurs marqués par un ou plusieurs objectifs d'intrusion de 0 à N (voir tableau 1).

	$Action_1$	$Action_2$	...	$Action_N$	$Objectifs$
F_1	$faux$	$faux$	...	$vrai$	1
F_2	$faux$	$faux$	...	$vrai$	0
...	...	...	...	...	...
F_n	$vrai$	$faux$	...	$faux$	$2, 4$

Tableau 1. *Données d'observations prétraitées*

En fait, les observations concernent tous les hôtes du réseau surveillé. Nous appliquons la procédure de prétraitement des données décrite ci-dessus pour chaque hôte individuellement et nous fusionnons à la fin les résultats obtenus dans un seul tableau. La procédure de prétraitement des données d'observation est résumée dans l'algorithme 1.

Nous allons maintenant illustrer cette première étape sur le premier scénario DARPA 2000. Ces données contiennent un trafic réseau brute capturé par un analyseur de trafic réseau pendant le plan d'attaque. Il nous faut maintenant décrire les actions de ce plan, ceci est fait avec l'aide d'un IDS (Snort[1]). Après l'analyse des données DARPA avec Snort, nous avons constaté que les alertes générées concernent les actions du tableau 2.

1. Snort est un système de détection d'intrusions, http ://www.snort.org

Algorithme 1: Prétraitement des données d'observations

Données: Historique des observations (alertes)
Result : Tableau de vecteurs;
début
  Grouper tous les objectifs d'intrusion dans une classe appelée "Objectifs-Intrusion";
  pour *chaque hôte* **faire**
    Trier les actions observées chronologiquement ensuite les regrouper en sous groupes, selon une certaine fenêtre de temps;
    si *un objectif d'intrusion est observé dans une fenêtre* **alors**
      Déplacer cette fenêtre à gauche jusqu'à ce qu'elle se termine sur cet objectif;
    Éttiqueter chaque sous groupe (vecteur) par le numéro correspondant à l'objectif d'intrusion;
  Arranger tous les vecteurs dans un seul tableau;
fin

Ces actions représentent l'ensemble des variables du RB naïf. Nous avons également observé une attaque DDoS réussie contre certains hôtes, donc cet objectif d'intrusion va représenter la classe du RB naïf.

Dans DARPA 2000, l'intrus a tenté de compromettre tous les hôtes du réseau. Il a obtenu trois hôtes compromis après l'étape 4 du scénario et il a lancé le DDoS contre la victime dans la dernière phase. L'attaque DDoS a été réalisée sur une période d'environ 3 heures sur 5 phases distinctes, donc nous allons prendre 3 heures comme une fenêtre de temps pour traiter les alertes de chaque machine individuellement. Le prétraitement des données DARPA 2000 a donné 44 vecteurs marqués avec DDoS lorsque la fenêtre concerne une attaque réussie, ou 0 lorsque la fenêtre concerne un trafic normal.

A_1 : icmp_ping
A_2 : rpc_sadmind_request
A_3 : sadmind_ping
A_4 : sadmind_root_query
A_5 : sadmind_bof
A_6 : icmp_reply
A_7 : telnet_info
A_8 : telnet_login_incorrect
A_9 : telnet_bad_login
A_{10} : rsh_root
A_{11} : icmp_port_unreachable

Tableau 2. *Actions onbservées dans DARPA 2000*

4.2. *Construction du RB naïf*

Nous construisons un RB naïf pour chaque objectif d'intrusion. La raison pour laquelle nous considérons un RB par objectif d'intrusion au lieu d'un seul RB avec une variable classe contenant tous les objectifs est que les objectifs d'intrusion ne sont pas exclusifs. Il peut arriver que deux objectifs d'intrusion différents $O1$ et $O2$ soient compromis simultanément, à savoir $P(O1) = P(O2) = 1$. En définissant un RB naïf par objectif d'intrusion, il est possible de représenter une telle situation. Toutefois, si un seul RB naïf est utilisé, nous allons avoir $P(O1) = P(O2) = 0,5$. Et s'il ya N objectifs qui sont compromis, alors nous ne pouvons pas représenter une telle situation et nous allons avoir $P(O_i) = \frac{1}{N}$, ce qui signifie que la probabilité de chaque objectif d'intrusion est faible. Maintenant, sur la base de ce constat, nous allons modi-

fier légèrement le tableau 1, en le fractionnant en plusieurs tableaux, chacun concerne un seul objectif d'intrusion. Plus précisément, pour chaque objectif d'intrusion nous remplaçons son numéro dans la colonne " objectifs" par " vrai", et les autres objectifs par "faux". Ainsi, nous obtenons un tableau pour chaque objectif d'intrusion.

Le tableau 3 montre les données prétraitées de l'objectif $O1$, la valeur "vrai" signifie que l'action/objectif a été observé(e) sur la fenêtre correspondante, la valeur "faux" signifie que l'action/objectif n'a pas été observé(e) sur la fenêtre correspondante.

	$Action_1$	$Action_2$	...	$Action_N$	O_1
W_1	$faux$	$faux$	...	$vrai$	$vrai$
W_2	$faux$	$faux$	...	$vrai$	$faux$
...	...	...	...	...	...
W_n	$vrai$	$faux$	...	$faux$	$faux$

Tableau 3. *Données prétraitées de l'objective $O1$*

La figure 2 montre le RB naïf du premier scénario DARPA 2000. La structure du réseau est déjà définie, il nous reste de calculer la distribution des probabilités.

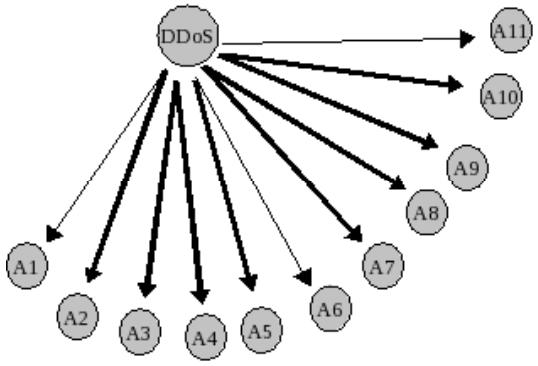

Figure 2. *RB naïf du premier scénario DARPA 2000*

Les données d'observations nous permettent d'estimer les distributions des probabilités conditionnelles qui peuvent être faites par un simple calcul de fréquences. Toutefois, lorsqu'une valeur d'un attribut ne se produit pas avec une valeur donnée de la classe, l'estimation du $P(A|C)$ produit une valeur nulle, et rend difficile l'étape de prédiction. Pour surmonter ce problème, nous utiliserons l'estimateur de Laplace. Compte tenu d'un facteur prédéfini f, s'il ya N instances de n exemple pour un problème de K valeurs, Laplace estime la probabilité par $(N + f)/(n + kf)$. Pour un problème binaire et avec $f = 1$, on obtient $(N + 1)/(n + 2)$ (Kohavi *et al.*, 1997).

Une fois les observations (alertes) sont obtenues et formatées comme dans le tableau 3, nous pouvons calculer la distribution des probabilités pour chaque variable.

La probabilité d'observer l'objectif d'intrusion $(P(class = vrai))$, et la probabilité de ne pas observer l'objectif d'intrusion $(P(class = faux))$ peuvent être calculées comme suit :

– $P(classe = X) = \frac{NB(classe=X)+1}{N+2}$ dont :

– $X \in \{vrai, faux\}$

– $NB(classe = X)$ nombre de lignes du tableau 3 dont $classe = X$

– N est la taille du tableau 3

La distribution des probabilités conditionnelles des variables dans le contexte de la classe peut être calculé comme suit :

– $P(action_j = Y \mid classe = X) = \frac{NB(action_j=Y \ et \ classe=X)+1}{NB(classe=X)+2}$ dont :

– $X, Y \in \{vrai, faux\}$

– $NB(action_j = Y \ et \ classe = X)$ nombre de lignes du table 3 dont $action_j = Y$ et $classe = X$

– $NB(classe = X)$ nombre de lignes du tableau 3 dont $classe = X$

La procédure de construction du RB naïf est résumée dans l'algorithme 2.

Algorithme 2: Construction du RB naïf

Données: Tableau des vecteurs
Result : RB naïfs;
début
  pour *chaque objectif d'intrusion* **faire**
    Remplacer son numéro dans le tableau des vecteurs par "vrai" et les autres par "faux";
    Calculer la distribution des probabilités pour le RB naïf correspondant;
fin

Les distributions des probabilités de l'objectif d'intrusion du premier scénario DARPA 2000 et les diférentes actions sont données dans les tableaux 4 et 5.

	Faux	Vrai
DDoS	91.3%	8.7%

Tableau 4. *Distributions des probabilités de l'objectif d'intrusion DDoS*

Le tableau 4 indique qu'à priori, il existe une faible probabilité qu'un DDoS soit observé.

4.3. *Prédiction des objectifs d'intrusion*

Notre objectif est de montrer comment inférer (prédire) les objectifs d'intrusion étant donné que certaines actions sont récemment observées.

Le but de l'inférence est d'estimer les valeurs des nœuds non observés, étant donné les valeurs des nœuds observés. Dans les RB naïfs, nous sommes intéressés à détermi-

			DDoS	
			Faux	Vrai
A_1	icmp_ping	Faux	4.65%	20%
		Vrai	95.35%	80%
A_2	rpc_sadmind_request	Faux	97.67%	20%
		Vrai	2.33%	80%
A_3	sadmind_ping	Faux	97.67%	20%
		Vrai	2.33%	80%
A_4	sadmind_root_query	Faux	97.67%	20%
		Vrai	2.33%	80%
A_5	sadmind_bof	Faux	97.67%	20%
		Vrai	2.33%	80%
A_6	icmp_reply	Faux	69.76%	20%
		Vrai	30.24%	80%
A_7	telnet_info	Faux	97.67%	20%
		Vrai	2.33%	80%
A_8	telnet_login_incorrect	Faux	97.67%	20%
		Vrai	2.33%	80%
A_9	telnet_bad_login	Faux	97.67%	20%
		Vrai	2.33%	80%
A_{10}	rsh_root	Faux	97.67%	20%
		Vrai	2.33%	80%
A_{11}	icmp_port_unreachable	Faux	55.81%	80%
		Vrai	44.19%	20%

Tableau 5. *Distribution des probabilités des actions*

ner les valeurs de la classe, étant donnés les valeurs de certaines variables observées, cela peut se faire par la formule de bayes :

$$P(classe = x|y) = \frac{P(classe = x).P(y|classe = x)}{P(y)},$$

où $classe$ est la variable non observée (dans notre cas, l'objectif d'intrusion) et y est l'évidence observée (dans notre cas, les actions observées). Quand les observations concernent plus d'une variable, cette formule peut être écrite comme suit :

$$\begin{aligned} P(classe = x|y_1, ..., y_n) &= \frac{P(classe = x, y_1, ..., y_n)}{P(y_1, ..., y_n)} \\ &= \frac{1}{\propto}.P(y_1, ..., y_n|classe = x).P(classe = x) \end{aligned}$$

Notons que $\propto$ est une constante qui peut être obtenue par normalisation. Maintenant,

$$\begin{aligned} P(classe = x, y_1, ..., y_n) &= P(y_1, y_2, ..., y_n|classe = x).P(classe = x) \\ &= P(y_1|y_2, ..., y_n, class = x). \\ &\quad P(y_2, ..., y_n|classe = x).P(classe = x) \end{aligned}$$

Rappelons qu'en RB naïf, par définition y_1, dans le contexte de la $class$, est independent de $y_2, ..., y_n$. D'où :

$$P(classe = x, y_1, ..., y_n) = P(y_1|classe = x) \cdot P(y_2, ..., y_n|classe = x).P(classe = x)$$

Et par itération non obtenons :

$$P(classe = x, y_1, ..., y_n) = P(y_1|classe = x).P(y_2|classe = x) \cdot \cdot P(y_n|classe = x).P(classe = x)$$

Dans notre contexte, le but de l'inférence est de calculer les nouvelles probabilités des objectifs d'intrusion étant donné que certaines actions sont observées. En présence d'une action observée, on distingue trois situations possibles :

1) Cette action appartient à un seul plan d'attaque. Dans ce cas, nous concentrons directement sur les autres actions du plan.

2) Cette action appartient à plusieurs plans d'attaque. Dans ce cas, nous concentrons sur les plans dont cette action influence le plus.

3) Cette action n'appartient à aucun RB naïf. Dans ce cas, la prédiction est seulement possible après la prochaine mise à jour du tableau 3.

Algorithme 3: Prédiction des objectifs d'intrusion

```
Données: Actions observées
Result : Prédiction des objectives d'intrusion ;
début
    Initialiser timeout;
    tant que timeout n'a pas expiré faire
        si une action A est observée alors
            pour Objectif O = O_1 to O_n faire
                si Influence(A,O) = Négative alors
                    Rien à faire;
                si Influence(A,O) = Positive alors
                    Concentrer sur cet objectif;
                si Influence(A,O) = Critique alors
                    Générer une alerte;
fin
```

Durant la détection, nous initialisons une variable, que nous notons "timeout", qui sera initialisée à 0 (zéro). Chaque nouvelle alerte générée engendera une nouvelle probabilitée de la classe. Selon l'influence de cette action sur les objectifs d'intrusion,

la probabilité de chaque objectif augmentera ou diminuera. Nous nous concentrons sur les plans d'attaques (RB naïfs), dans lesquels la probabilité de l'objectif d'intrusion a augmenté.

Après chaque mise à jour, nous vérifions la nouvelle probabilité d'atteindre chaque objectif d'intrusion. Si la nouvelle probabilité dépasse un certain seuil, une alarme est générée. Si aucun objectif d'intrusion ne dépasse le seuil, nous attendons la prochaine alerte. Lorsque le timeout expire et la probabilité de l'objectif d'intrusion n'a pas dépassé le seuil, nous pouvons confirmer qu'aucun plan d'attaque n'est en place. Après l'expiration du timeout, nous reinitialisons la phase de détection. La procédure de prédiction est résumée dans l'algorithme 3

Voyons maintenant comment chaque action du premier scénario DARPA 2000 influence l'objectif d'intrusion DDoS. Á priori le RB naïf du DARPA 2000 (figure 2) n'indique rien sur le plan d'attaque, mais après l'application d'un simple calcul d'influence entre les variables et la classe (la classe est l'objectif d'intrusion), nous pouvons clairement identifier les actions impliquées dans le plan d'attaque.

Le tableau 6 montre l'influence de chaque action représentée par la nouvelle probabilité de l'objectif d'intrusion. Les actions A_3, A_4 , A_5, A_7 , A_8, A_9 et A_{10} ont une influence critique sur l'objectif d'intrusion, car la probabilité d'atteindre l'objectif d'intrusion, sachant chacune de ces actions, dépasse 50% (voir tableau 6). Les actions A_2 et A_6 ont une influence positive sur l'objectif d'intrusion, car la probabilité d'atteindre l'objectif d'intrusion, sachant chacune de ces actions, augmente sans dépasser 50%. Les autres actions ont une influence négative sur l'objectif d'intrusion, car la probabilité d'atteindre l'objectif d'intrusion, sachant chacune de ces actions, diminue.

	$P(DDoS\|Aj)$
A_1	7.4%
A_2	29.1%
A_3	76.6%
A_4	76.6%
A_5	76.6%
A_6	20.1%
A_7	76.6%
A_8	76.6%
A_9	76.6%
A_{10}	76.6%
A_{11}	4.1%

Tableau 6. *Influence des actions sur le DDoS*

Cette analyse ne concerne que la première étape de la prédiction, à savoir si une seule action est observée. Maintenant, nous allons voir comment effectuer cette analyse sur la base des alertes rapportées.

Nous avons illustré l'influence des actions individuellement. Maintenant, nous allons illustrer la phase de prédiction avec deux scénarios réels, extraits des données DARPA 2000. Ces deux scénarios représentent respectivement un cas de succés et un cas d'échec de l'attaque DDoS contre deux hôtes distincts. Ces deux scénarios ont été retirées de l'étape d'apprentissage, à savoir

	$P(DDoS\|Aj)$
A_1	7.4%
A_1, A_2	25.6%
A_1, A_2, A_3	92.2% $\Leftarrow$
A_1, A_2, A_3, A_4	99.8%
A_1, A_2, A_3, A_4, A_5	100%
$A_1, A_2, A_3, A_4, A_5, A_6$	100%
$A_1, A_2, A_3, A_4, A_5, A_6, A_7$	100%
$A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8$	100%
$A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9$	100%
$A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}$	100%

Tableau 7. *Cas de succés*

le prétraitement des données et la construction du RB naïf pour les utiliser dans la phase de prédiction. Ces deux scénarios seront utilisés pour tester notre approche.

La probabilité, avant de recevoir aucune alerte, que l'objectif d'intrusion DDoS soit atteint est 8,7% (voir tableau 4). Après avoir rejouer le premier scénario, Snort a détecté cet ensemble d'actions {A1, A2, A3, A4, A5, A6, A7, A8, A9, A10 }, qui sont triées par ordre chronologique. Après avoir généré chaque alerte, nous avons mis à jour ces observations dans le RB naïf et nous avons inféré la nouvelle probabilité d'atteindre le DDoS (voir tableau 7). Selon les nouvelles probabilités, il est clair qu'après la génération de l'alerte A3, nous pouvons confirmer que le DDoS peut être atteint directement, sans atteindre l'expiration du délai. Une alerte sera donc générée. Notons que cette action (A3) a une grande influence sur l'objectif d'intrusion, parce que tous les hôtes exécutant Sadmind dans les données DARPA 2000 ont été compromis.

Après avoir rejoué le deuxième scénario, Snort a détecté cet ensemble d'actions {A1, A2, A6, A11 }, qui sont triées par ordre chronologique. Après avoir généré chaque alerte, nous avons mis à jour ces observations dans le RB naïf et nous avons inféré la nouvelle probabilité du DDoS (voir le tableau 8). Après avoir généré A11, nous n'avons pas observé d'autres actions jusqu'à l'expiration du délai d'attente. Une fois que le délai est expiré, nous avons constaté que la probabilité d'atteindre l'objectif d'intrusion n'a pas dépassé le seuil, donc nous pouvons confirmer que le DDoS ne peut pas être atteint (le trafic est normal) et nous redémarrons la phase de détection.

	$P(DDoS\|Aj)$
A_1	7.4%
A_1, A_2	25.6%
A_1, A_2, A_6	47.6%
A_1, A_2, A_6, A_{11}	29.2%

Tableau 8. *Cas d'échec*

5. Traveaux précédents

Les réseaux bayésiens ont été utilisés dans la détection d'intrusion dans plusieurs travaux de cherche, tels que : Classificateur pour la détection d'intrusion (Axelsson, 2004)(Amor *et al.*, 2004)(Kang *et al.*, 2005)(Krügel *et al.*, 2003)(Puttini *et al.*, 2003), la cybercriminalité (Abouzakhar *et al.*, 2003), Reconnaissance de plan d'attaque (Geib *et al.*, 2001)(Qin *et al.*, 2004) et Détection d'intrusions distribuée et multi-agents (Burroughs *et al.*, 2002)(Gowadia *et al.*, 2005)(Scott, 2004), etc.

Axelsson (Axelsson, 2004) a proposé un système de détection basé sur les statistiques de Bayes combiné avec un composant de visualisation, afin de palier aux faibles taux de détection et le taux élevé de fausses alarmes. Cette approche est basée sur le principe de filtrage bayésien, exactement comme le filtrage des spams dans le courrier électronique. Elle permet au système de faire la différence entre les accès normaux et malicieux.

Abouzakhar et al (Abouzakhar *et al.*, 2003) ont proposé une approche d'apprentissage des réseaux bayésiens pour la détection de la cybercriminalité, afin de détecter les attaques distribuées le plus tôt possible.

Dans (Scott, 2004) Scott a décrit un paradigme pour la conception d'un système de détection d'intrusions réseau basé sur des modèles stochastiques. Le principe est de baser la détection d'intrusions sur les modèles stochastiques de l'utilisateur combiné au comportement des intrus en utilisant le théorème de Bayes.

Plus récemment, Gowdia et al (Gowadia *et al.*, 2005) ont mis au point un système de détection d'intrusions probabiliste muli-agents. Ce système est une architecture coopérative multi-agents dans laquelle des agents autonomes peuvent effectuer des taches spécifiques de détection d'intrusion et collaborer avec d'autres agents en partageant leurs croiyances sur un réseau bayésien partagé, fournie par un expert.

Tous les travaux ci-dessus appliquent les réseaux bayésiens à la détection d'intrusions, mais aucun de ces travaux n'a utilisé les réseaux bayésiens pour détecter des attaques coordonnées. Maintenant, parmi les travaux existants, celui de Qin et Lee (Qin *et al.*, 2004) est le plus proche de notre approche.

Qin et Lee (Qin *et al.*, 2004) ont proposé une approche pour la reconnaissance et la prédiction des plans d'attaque en utilisant des réseaux de causalité. Dans cette approche, les auteurs utilisent des arbres de décision pour définir une bibliothèque de plans d'attaque pour corréler les alertes. Ils transforment ensuite ces arbres en réseaux bayésiens sur lequel ils peuvent affecter une distribution de probabilité en intégrant les domaines de connaissances nécessaires, pour enfin évaluer le risque des objectifs d'intrusion et de prédire les futures attaques.

Il est clair que la principale différence avec notre approche est que les arbres d'attaques doivent être explicitement définies par un expert (Qin *et al.*, 2004), alors que dans notre approche, elles sont obtenues automatiquement (nous n'avons pas besoin de déterminer a priori l'ensemble des actions impliquées dans les scénarios). Ceci est un avantage important de notre approche. Notre approche est plus facile à mettre en oeuvre et n'implique pas une grande contribution des connaissances d'experts. L'administrateur n'a qu'à déterminer les objectifs d'intrusions à protéger et mémoriser quand ces objectifs ont été compromis dans l'historique des observations. De plus, notre approche filtre implicitement les fausses alarmes. Chaque alerte influençant négativement l'objective d'intrusion sera considérée comme non pertinente à cet objectif.

6. Conclusion

Dans cet article, nous avons proposé une nouvelle méthode de corrélation d'alertes basées sur les RB naïfs. Notre approche utilise l'historique des observation pour construire un Rb naïf pour chaque objectif d'intrusion. Pendant l'étape de détection, chaque action observée se traduit par une évidence qui mit à jour chaque RB naïf. Selon le degré d'influence de cette action, la probabilité de chaque objectif d'intrusion change positivement ou négativement.

Notre approche a pour avantage de rendre la prédiction des plans d'attaque plus facile grâce à la simplicité et l'efficacité des RB naïfs. Elle tire profit des données disponibles, et n'implique qu'une légère contribution des connaissance d'experts pour déterminer les objectifs d'intrusion. En plus, les actions impliquées dans les plans d'attaque peuvent être identifiées et les fausses alarmes sont implicitement filtrées en se concentrant sur les actions pertinentes.

Contrairement aux approches existantes, les scénarios d'attaque ne sont pas explicitement fournis par des experts, mais ils sont calculés automatiquement à partir des données d'observations.

7. Bibliographie

Abouzakhar N. S., Gani A., Manson G., Abuitbel M., King D., « Bayesian Learning Networks Approach to Cybercrime Detection », *the 2003 PostGraduate Networking Conference*, 2003.

Amor N. B., Benferhat S., Elouedi Z., « Naive Bayes vs decision trees in intrusion detection systems », *SAC*, p. 420-424, 2004.

Axelsson S., « Combining a Bayesian Classifier with Visualisation : Understanding the IDS », *VizSEC/DMSEC-04 ACM*, p. 99-108, 2004.

Benferhat S., Autrel F., Cuppens F., « Enhanced Correlation in an Intrusion Detection Process », *MMM-ACNS*, p. 157-170, 2003.

Burroughs D. J., Wilson L. F., Cybenko G. V., « Analysis of Distributed Intrusion Detection Systems Using Bayesian Methods », *21th IEEE International Conference on Performance, Computing, and Communications*, p. 329-334, 2002.

Cuppens F., « Managing Alerts in a Multi-Intrusion Detection Environment », *ACSAC*, p. 22-31, 2001.

Cuppens F., Miège A., « Alert Correlation in a Cooperative Intrusion Detection Framework », *IEEE Symposium on Security and Privacy*, p. 202-215, 2002.

Dain O., Cunningham R. K., « Fusing a heterogeneous alert stream into scenario », *ACM Workshop on Data Mining for Security Application*, p. 1-13, 2001.

DARPA-2000, *http ://www.ll.mit.edu/IST/ideval/data/data_index.html*, 2000.

Debar H., Wespi A., « Aggregation and Correlation of Intrusion-Detection Alerts », *Recent Advances in Intrusion Detection*, p. 85-103, 2001.

Duda R. O., Stork D. G., Hart P. E., *Pattern Classification*, Wiley, 2000.

Friedman N., Goldszmidt M., « Building classifiers using bayesian networks », *AAAI*, 1996.

Geib C. W., Goldman R. P., « Plan Recognition in Intrusion Detection Systems », *DISCEX*, vol. 1, p. 46-55, 2001.

Gowadia V., Farkas C., Valtorta M., « PAID : A Probabilistic Agent-Based Intrusion Detection system », *Computers & Security*, vol. 24, n° 7, p. 529-545, 2005.

Jensen F. V., *Introduction to Bayesian networks*, UCL Press, London, 1996.

Julisch K., « Mining Alarm Clusters to Improve Alarm Handling Efficiency », *ACSAC*, p. 12-21, 2001.

Kang D.-K., Fuller D., Honavar V., « Learning Classifiers for Misuse and Anomaly Detection Using a Bag of System Calls Representation », *IEEE Workshop on Information Assurance and Security*, p. 118-125, 2005.

Kohavi R., Becker B., Sommerfield D., « Improving simple Bayes », *European Conference on Machine Learning*, 1997.

Krügel C., Mutz D., Robertson W. K., Valeur F., « Bayesian Event Classification for Intrusion Detection », *ACSAC*, p. 14-23, 2003.

Ning P., Cui Y., Reeves D. S., « Analyzing Intensive Intrusion Alerts via Correlation », *RAID*, p. 74-94, 2002.

Pearl J., « Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference », *Artif. Intell.*, vol. 48, n° 1, p. 117-124, 1991.

Puttini R., Marrakchi Z., Mè L., « A Bayesian Classification Model for Real-Time Intrusion Detection », *22nd International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, vol. 659, p. 150-162, 2003.

Qin X., Lee W., « Attack Plan Recognition and Prediction Using Causal Networks », *ACSAC*, p. 370-379, 2004.

Scott L. S., « A Bayesian paradigm for designing intrusion detection systems », *Computational Statistics & Data Analysis*, Elsevier, p. 69-83, 2004.

Shachter R. D., Peot M. A., « Decision Making Using Probabilistic Inference Methods », *UAI*, p. 276-283, 1992.

Steven J. T., Karm L., « A requires/provides model for computer attacks », *New Security Paradigms Workshop*, p. 31-38, 2000.

Valdes A., Skinner K., « Probabilistic Alert Correlation », *Recent Advances in Intrusion Detection*, p. 54-68, 2001.