# Réseaux bayésiens pour le filtrage d'alarmes dans les systèmes de détection d'intrusions

Ahmad Faour[1,2] ahmad.faour@ul.edu.lb
Philippe Leray[1] philippe.leray@insa-rouen.fr
Cédric Foll[1,3] cedric.foll@ac-rouen.fr

[1] Laboratoire PSI - FRE CNRS 2645, INSA Rouen, France
[2] Laboratoire LPM, Université Libanaise, Beyrouth, Liban
[3] Rectorat de Rouen, France

## 1 Introduction

La détection des tentatives d'attaques sur un réseau est une problématique très importante dans le domaine de la sécurité informatique. Les NIDS (*Network Intrusion Detection Systems*), systèmes de détection d'intrusions, génèrent tellement d'alertes sur un réseau qu'il en devient très difficile de déterminer celles générées par une attaque réelle. L'utilisation d'outils de raisonnement probabiliste comme les réseaux bayésiens (RB) peut être efficace pour détecter les problèmes réels. Nous allons donc tout d'abord présenter les systèmes de détection d'intrusions et leurs limites puis passer brièvement en revue l'application de méthodes d'apprentissage à cette problématique. Nous décrirons enfin notre architecture de filtrage d'alarmes issues de NIDS.

## 2 Systèmes de détection d'intrusions

Les firewalls utilisés sur les réseaux TCP/IP fonctionnent sur l'analyse des couches IP et TCP/UDP/ICMP, pour déterminer quelles sont les machines impliquées dans la connexion et à quel service la connexion s'adresse. Ce genre d'approche, bien que nécessaire, se révèle insuffisant dans bien des cas (Chambet, 2002). Il faut donc pousser plus loin l'analyse en examinant aussi les couches réseaux supérieures. Cette tâche, plus difficile, est dévolue aux NIDS. Ces logiciels fonctionnent le plus souvent par signatures, sur le même principe que les anti-virus (Zimmermann et al., 2002), en répertoriant les attaques connues. Une alarme est donc générée à chaque fois qu'une trame réseau ressemble à une des attaques répertoriées. Lorsqu'un nouvel exploit (tentative d'intrusion réussie) est répertorié, une signature adaptée sera ajoutée à la base de signatures. Cette approche est souvent utilisée conjointement avec une approche statistique dans laquelle le NIDS détermine d'abord un profil type du réseau (nombre de paquets échangés, volume des flux, nombre de connections, etc.) et alarme ensuite l'administrateur lorsque le trafic courant dévie de ce profil. Malheureusement, les NIDS émettent généralement une quantité importante d'alarmes que l'administrateur n'est pas capable d'interpréter rapidement.

Depuis (Denning, 1987), les approches à base d'apprentissage statistique proposées pour la détection d'intrusion peuvent être classées en deux types : les méthodes essayant d'opérer avec les mêmes informations que les NIDS classiques (analyse de données réseaux), et celles opérant à partir de données comportementales de plus haut niveau (fichiers de logs de certaines applications ou du système).

Parmi les méthodes du premier type, citons l'utilisation de statistiques simples (Bykova et al., 2001) ou de réseaux bayésiens (Abouzakhar et al., 2003; Ben Amor et al., 2003). (Hood and Ji, 1997) applique des réseaux bayésiens hiérarchiques latents pour l'analyse des paquets et une modélisation auto-régressive pour l'analyse temporelle des alertes tandis que (Valdes and Skinner, 2000) utilise directement des réseaux bayésiens temporels pour modéliser les attaques et leur évolution temporelle.

Les RB sont souvent employés pour des analyses de plus haut niveau (ressources systèmes, logs utilisateurs) (Bronstein et al., 2001; Sebyala et al., 2002; Kruegel et al., 2003; Puttini et al., 2003; Scott, 2004). Pour l'analyse temporelle d'attaques, (Seleznyov, 2000) utilise un modèle graphique probabiliste (arbre temporel). (Cho, 2002) met en œuvre des modèles de Markov cachés, modèles graphiques probabilistes très proches des RB temporels. (Burroughs et al., 2002) utilise une règle de décision bayésienne très simple sur les logs de NIDS.

# 3 Notre approche

L'approche que nous proposons (fig. 1-a) se situe à mi-chemin entre les deux familles de méthodes précédentes, en travaillant à partir des fichiers de logs d'un logiciel de NIDS standard. Cette approche se décompose en trois étapes (prétraitement temporel, spatial, classification).

**Prétraitement temporel**
Une attaque est souvent caractérisée par une série d'événements consécutifs tentant de violer la politique de sécurité d'une machine. Nous allons essayer de détecter de tels comportements en nous plaçant dans un mode de détection pseudo-réel, *i.e.* entre le temps réel (une détection à chaque mesure sur le réseau) et l'analyse *off-line* (pouvant être effectuée une fois par jour par exemple). Pour cela, nous effectuons une synthèse des différents types d'alarmes générées par le NIDS pendant une fenêtre de temps mobile de 1 heure avec une réactualisation toutes les 20 minutes. Ces valeurs ont été déterminées par un expert pour avoir un bon compromis entre la durée minimale

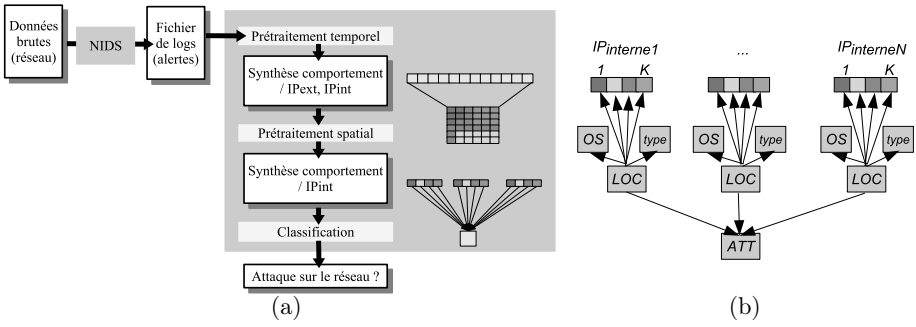


FIG. 1 – *(a) Architecture générale – (b) Modélisation modulaire de l'étape de classification.*

nécessaire pour détecter les scénarii potentiels d'attaque et une durée maximale au delà de laquelle le système est noyé par les alertes. Nous comptabilisons donc, pour la fenêtre de temps considérée, le nombre d'alarmes de chaque type pour chaque valeur du couple $(IP_{interne}, IP_{externe})$, sans tenir compte de la valeur du port externe (jugé non significatif par notre expert) ni du port interne ou du protocole (très corrélés avec le type d'alarme généré par le NIDS). Ce nombre d'alarmes est ensuite "normalisé" pour être ramené au trafic moyen pour chaque machine $IP_{interne}$.

**Prétraitement spatial**
Nous partons du principe que le nombre d'alarmes (normalisé) de chaque type obtenu précédemment est représentatif du comportement de chaque machine $IP_{externe}$ à destination de chaque machine interne. Nous supposons aussi que ce comportement peut être similaire pour plusieurs machines externes (qui tenteraient le même genre d'attaque vers une même machine interne), ou à destination de plusieurs machines internes (une même attaque pourrait être dirigée vers plusieurs machines). Nous utilisons donc une technique de clustering classique (projection dans une carte auto-organisatrice de Kohonen puis regroupement éventuel des cases de cette carte par l'algorithme des K-moyennes) pour regrouper ces comportements en $K$ comportements-types.

**Classification**
Le nombre de comportements de chaque type détectés à destination d'une $IP_{interne}$ est censé être représentatif des différents types d'attaques potentielles visant cette machine pendant une fenêtre de temps fixée. Nous utilisons ces informations dans cette dernière phase pour tenter de déterminer si le réseau a été réellement attaqué $(ATT = vrai|faux)$. Pour cette tâche de classification, nous proposons de tirer le meilleur parti des réseaux bayésiens en étudiant deux types de modélisation.

Le premier type de modélisation, assez sommaire, a donné d'excellents résultats dans de nombreux domaines. Nous considérons le vecteur caractéristique de toutes les machines du réseau pour déterminer directement la valeur de $ATT$. Parmi les structures de RB généralement utilisées en classification (Leray and Francois, 2004; Naïm et al., 2004), nous avons comparé une structure naïve (NB) et trois structures déterminées à partir de données d'apprentissage de notre système : une structure naïve augmentée par un arbre (TANB), un arbre optimal (MWST) et un réseau de type Multinet. Le deuxième type de modélisation, modulaire (fig. 1-b), incorpore dans la structure du RB des informations spécifiques à la tâche à résoudre. Nous pouvons prendre en compte la topologie du réseau informatique concerné, le système d'exploitation et le rôle de chaque machine (web, mail ...).

**Premières expérimentations**
Notre fichier de logs du NIDS SNORT comprend 32031 alertes générées (de 406 types différents) sur une durée de 20 jours. Ces alertes correspondent à 4638 machines extérieures essayant de se connecter sur 288 IP internes. Elles correspondent à 15 attaques réelles dont la durée varie de quelques minutes à plusieurs jours. Nos premières expérimentations de modélisation "sommaire" semblent prometteuses, avec un *Hit Rate* de 100% (toutes les *vraies* attaques sont reconnues) et un *False Alarm* proche de 26% (3/4 des alarmes ne correspondant pas à des vraies attaques sont supprimées).

# 4 Conclusion et perspectives

Nous avons proposé une architecture de filtrage d'alarmes dont le module de classification final met en œuvre des réseaux bayésiens plus ou moins complexes. Deux approches sont proposées, l'une raisonnant avec le moins possible de connaissances *a priori* sur la tâche à résoudre, l'autre (actuellement en phase de test) essayant de prendre en compte les spécificités du réseau (topologie, caractéristiques des machines). Nous comptons aussi privilégier l'aspect évolutif de notre approche en adaptant chaque partie de notre architecture à l'apparition de nouveaux types d'alarmes, de nouvelles attaques ou de nouvelles machines dans le réseau.

# Références

Abouzakhar, N., Gani, A., Manson, G. et King, D. (2003). Bayesian learning networks approach to cybercrime detection. In PGNET 2003, John Moore University, Liverpool, 16-17 June 2003.

Ben Amor, N., Benferhat, S. et Elouedi, Z. (2003). Naive bayesian networks in intrusion detection systems. In Proceedings of the ECML03 Workshop on Probabilistic Graphical Models for Classification, pp 11–24.

Bronstein, A., Cohen, I., Das, J., Duro, M., Friedrich, R., Kleyner, G., Mueller, M. et Singhal, S. (2001). Self-aware services : Using bayesian networks for detecting anomalies in internet-based services. Technical report, HP Labs.

Burroughs, D. J., Wilson, L. F. et Cybenko, G. V. (2002). Analysis of distributed intrusion detection systems using bayesian methods. In Proceedings of IEEE International Performance Computing and Communication Conference.

Bykova, M., Ostermann, S. et Tjaden, B. (2001). Detecting network intrusions via a statistical analysis of network packet characteristics. In Proceedings of the 33rd Southeastern Symposium on System Theory.

Chambet, P. (2002). Firewalls et applications web : architecture et sécurisation - pourquoi les firewalls sont impuissants face aux attaques web. Linux Magazine.

Cho, S. (2002). Incorporating soft computing techniques into a probabilistic intrusion detection system. IEEE Transactions on Systems, Man, and Cybernetics, 32(2) :154–160.

Denning, D. E. (1987). An intrusion-detection model. IEEE Trans. Softw. Eng., 13(2) :222–232.

Hood, C. S. et Ji, C. (1997). Proactive network fault detection. In INFOCOM '97, page 1147. IEEE Computer Society.

Kruegel, C., Mutz, D., Robertson, W. et Valeur, F. (2003). Bayesian event classification for intrusion detection. In 19th Annual Computer Security Applications Conference.

Leray, P. et Francois, O. (2004). Réseaux bayésiens pour la classification – méthodologie et illustration dans le cadre du diagnostic médical. Revue d'Intelligence Artificielle, 18/2004 :169–193.

Naïm, P., Wuillemin, P.-H., Leray, P., Pourret, O. et Becker, A. (2004). Réseaux bayésiens. Eyrolles.

Puttini, R. S., Marrakchi, Z. et Mé, L. (2003). A bayesian classification model for real-time intrusion detection. AIP International Conference, 659(1) :pp. 150–162.

Scott, S. L. (2004). A bayesian paradigm for designing intrusion detection system. Computational Statistics and Data Analysis (special issue on network intrusion detection), 45 :69–83.

Sebyala, A. A., Olukemi, T. et Sacks, L. (2002). Active platform security through intrusion detection using naïve bayesian network for anomaly detection. In London Communications Symposium.

Seleznyov, A. (2000). Temporal-probabilistic network approach for anomaly intrusion. In 12th Annual Computer Security Incident Handling Conference, Chicago.

Valdes, A. et Skinner, K. (2000). Adaptive, model-based monitoring for cyber attack detection. In Debar, H., Me, L. et Wu, F., editors, Recent Advances in Intrusion Detection (RAID 2000), number 1907 in Lecture Notes in Computer Science, pp 80–92, Springer-Verlag.

Zimmermann, J., Mé, L. et Bidan, C. (2002). Introducing reference flow control for intrusion detection at the OS level. In Proceedings of RAID 2002. Springer Verlag.