Diagnostic multi-sources adaptatif
Application à la détection d'intrusion dans des serveurs Web

Thomas Guyet*, Wei Wang*,**
René Quiniou*, Marie-Odile Cordier*

*INRIA/IRISA - Université Rennes 1
{thomas.guyet, rene.quiniou, marie-odile.cordier}@irisa.fr,
http://www.irisa.fr/dream/Pages_Pros/Thomas.Guyet/
**Sophia Antipolis/INRIA
wwangemail@gmail.fr

Résumé. Le but d'un système adaptatif de diagnostic est de surveiller et diagnostiquer un système tout en s'adaptant à son évolution. Ceci passe par l'adaptation des diagnostiqueurs qui précisent ou enrichissent leur propre modèle pour suivre au mieux le système au fil du temps. Pour détecter les besoins d'adaptation, nous proposons un cadre de diagnostic multi-sources s'inspirant de la fusion d'information. Des connaissances fournies par le concepteur sur des relations attendues entre les diagnostiqueurs mono-source forment un méta-modèle du diagnostic. La compatibilité des résultats du diagnostic avec le méta-modèle est vérifiée en ligne. Lorsqu'une de ces relations n'est pas vérifiée, les diagnostiqueurs concernés sont modifiés.

Nous appliquons cette approche à la conception d'un système adaptatif de détection d'intrusion à partir d'un flux de connexions à un serveur Web. Les évaluations du système mettent en évidence sa capacité à améliorer la détection des intrusions connues et à découvrir de nouveaux types d'attaque.

1 Introduction

Les systèmes automatiques de surveillance sont de plus en plus répandus. Ils ont pour tâche d'émettre des alarmes lors de dysfonctionnements de systèmes aussi variés que les patients en unités de soins intensifs, les systèmes physiques (*e.g.* voitures, machines industrielles) ou informatiques (*e.g.* les serveurs Web). Si les données disponibles sur le fonctionnement des systèmes surveillés sont de plus en plus riches, et si les techniques de monitoring sont de plus en plus performantes, l'adaptation en ligne du monitoring reste un défi important pour assurer une surveillance précise, robuste, en continu et ne nécessitant que peu d'intervention humaine. En particulier, l'adaptation en ligne de ces systèmes doit permettre de :

– faciliter l'installation d'un système de surveillance en le laissant automatiquement s'adapter aux conditions particulières de son utilisation (*e.g.* adaptation aux caractéristiques physiologiques d'un patient),

– assurer un fonctionnement en continu malgré les évolutions naturelles du système (*e.g.* adaptation à l'usure d'un système physique) ou du contexte (*e.g.* adaptation à des changements de température),
– intégrer dynamiquement de nouvelles "situations" (*e.g.* adaptation à de nouveaux types d'intrusion dans des serveurs Web).

Nous sommes en particulier motivés par la conception d'un système de détection d'intrusion dans des serveurs Web. Les modifications des contenus d'un serveur Web ainsi que l'apparition fréquente de nouveaux types d'intrusion justifient le besoin d'adaptation automatique d'un système séparant les requêtes intrusives de celles destinées à être traitées normalement.

Si de tels systèmes adaptatifs n'existent pas à l'heure actuelle, c'est que, outre le problème du temps disponible pour réaliser dynamiquement les adaptations nécessaires, l'adaptation en ligne soulève deux problèmes majeurs : (1) le choix des informations utilisées pour l'adaptation, et (2) le choix de la direction de l'adaptation à mener. Lorsqu'on automatise la construction hors-ligne des modèles d'un système dans le but de les diagnostiquer en ligne, on dispose de données annotées qui permettent de connaître quels sont les exemples et les contre-exemples des situations qui doivent être modélisées. De plus, les différents modèles peuvent être confrontés les uns aux autres pour identifier les meilleurs. Lorsque l'adaptation doit être réalisée en ligne, d'une part, le diagnostiqueur ne dispose pas d'annotations sur la situation courante puisque il doit lui-même décider de la nature de la situation courante, et d'autre part, il ne dispose pas des ressources permettant de retenir toutes les données et les différents modèles pour les confronter.

Dans cet article, nous nous intéressons à l'utilisation d'un méta-modèle de diagnostic, défini à partir de connaissances *a priori*, pour décider dynamiquement des adaptations à mener. Un méta-modèle de diagnostic, tel que celui de De Kleer (2007), modélise des comportements attendus du diagnostiqueur et permet de détecter en ligne les manquements et les défaillances des modèles utilisés pour le diagnostic. Nous proposons une méthode de diagnostic multi-sources qui utilise plusieurs sources d'informations sur l'état d'un système pour en déterminer l'état global. Par exemple, Fromont et al. (2005) utilisent des signaux provenant de plusieurs capteurs pour du monitoring cardiaque. Chaque source est traitée par un diagnostiqueur mono-source (DMoS). Les diagnostics fournis par les DMoS sont progressivement fusionnés pour fournir un diagnostic global en suivant un schéma défini par le méta-modèle et leur compatibilité avec les contraintes exprimées dans le méta-modèle est vérifiée pour identifier les besoins d'adaptation. Ces besoins d'adaptation sont utilisés individuellement par les DMoS pour qu'ils réalisent leur propre adaptation.

2 Diagnostic multi-sources adaptatif

2.1 Diagnostic et décision de diagnostic

Notre architecture de diagnostic multi-sources reprend certains formalismes introduits dans le domaine de la fusion d'information que nous rappelons ici pour donner nos définitions de **diagnostic** et de **décision de diagnostic**. Parmi les techniques de fusion d'information existantes (réseaux de neurones, réseaux bayésiens, théorie possibiliste, ...) nous avons choisi d'utiliser la théorie de Dempster-Shafer (1976) (DS), pour sa capacité à intégrer la méconnaissance dans la prise de décision.

L'ensemble des **décisions de diagnostic possibles**, noté Ω et appelé espace de discernement dans la théorie de DS, représente les états possibles du système à diagnostiquer. Un **diagnostic** est une distribution normalisée de masses d'évidence sur les parties de Ω. C'est-à-dire qu'un diagnostic est une fonction :

$$m : 2^{\Omega} \mapsto [0,1], \sum_{A \in 2^{\Omega}} m(A) = 1$$

La distribution m associe des degrés de croyance (Bel_m) et de plausibilité (Pl_m) aux parties de Ω :

$$\forall A \in 2^{\Omega},\ Bel_m(A) = \sum_{B \subset A} m(B), \quad Pl_m(A) = \sum_{B \cap A} m(B).$$

Par définition, la **décision de diagnostic** est la partie de Ω qui a le plus fort degré de croyance, *i.e.* $\arg\max_{A \in 2^{\Omega}} (Bel_m(A))$.

2.2 Éléments du diagnostic multi-sources adaptatif

Avant d'introduire les mécanismes qui réalisent le diagnostic et l'adaptation, on introduit dans cette section les trois éléments principaux de notre cadre de diagnostic multi-sources adaptatif : les diagnostiqueurs mono-source, les diagnostiqueurs multi-sources et le méta-modèle de diagnostic.

2.2.1 Diagnostiqueur mono-source (DMoS)

Un diagnostiqueur mono-source (DMoS) prend en entrée les observations sur le système à diagnostiquer et construit en ligne un *diagnostic élémentaire* en appliquant, d'une part, une **fonction d'extraction d'information** et, d'autre part, une **fonction de diagnostic**. À partir des observations, la fonction d'extraction d'information construit une information qui pourra être utilisée par la fonction de diagnostic. Son rôle est de permettre, d'une part, de focaliser le diagnostiqueur sur une partie des observations (filtrage et concentration de l'information) et, d'autre part, de structurer celle-ci. La fonction de diagnostic construit le diagnostic élémentaire à partir des informations extraites.

Un DMoS est également doté d'une **fonction d'adaptation** qui lui est propre. Elle prend comme paramètre une **proposition d'adaptation** formée (1) d'observations o, en pratique il s'agit des observations en cours d'analyse, et (2) d'une décision de diagnostic d qui est attendue pour ces observations. L'adaptation modifie la fonction de diagnostic de sorte que celle-ci fournisse un diagnostic pour o "plus proche" de d qu'elle ne le faisait auparavant.

2.2.2 Diagnostiqueurs multi-sources (DMuS)

Un diagnostiqueur multi-sources (DMuS) prend en entrée plusieurs diagnostics : diagnostics élémentaires ou diagnostics provenant d'autres DMuS, dits *intermédiaires*. Il propose en sortie un nouveau diagnostic intermédiaire obtenu par la fusion des diagnostics d'entrée. Contrairement aux DMoS, les DMuS n'ont pas de fonction d'adaptation.

La cadre théorique de la fusion d'information de DS définit la règle de fusion des diagnostics, notée $\oplus$, que nous utilisons. Elle consiste à construire un diagnostic m à partir de deux[1] diagnostics définis par leurs distributions de masses m_1 et m_2. Pour toute partie A de l'espace de discernement, sa masse est calculée par la formule :

$$m(A) = (m_1 \oplus m_2)(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)}.$$

2.2.3 Méta-modèle du diagnostic

Un méta-modèle de diagnostic définit les DMuS à partir des DMoS disponibles en précisant (1) les sources des diagnostics en entrée des DMuS et (2) les contraintes associées à chaque DMuS.

Le méta-modèle peut être représenté sous la forme d'un arbre dans lequel les nœuds sont des DMuS et les feuilles sont des DMoS. Les sous-branches d'un nœud D précisent les diagnostics qui sont utilisés en entrée par le DMuS D et la branche supérieure figure le diagnostic de sortie de D. Le diagnostic à la sortie du DMuS situé à la racine de l'arbre donne le *diagnostic global*.

Une **contrainte** définit une relation systématiquement attendue entre les diagnostics en entrée et en sortie du DMuS. Nous verrons en Section 2.3.2 comment ces contraintes sont utilisées pour proposer des adaptations. Dans cet article, on définit un seul type de contrainte : la *concordance* du diagnostic de sortie du DMuS avec chacun de ses diagnostics d'entrée. Si on note $D(o)$ le diagnostic construit par un diagnostiqueur D pour des observations o, alors D et D' *concordent* si et seulement si ils décident de la même décision de diagnostic (*i.e.* $\arg\max_{A \in 2^\Omega} Bel_{D(o)}(A) = \arg\max_{A \in 2^\Omega} Bel_{D'(o)}(A)$). Dans la représentation du méta-modèle sous la forme d'un arbre, les nœuds sont décorés avec la contrainte à laquelle le DMuS correspondant est associé.

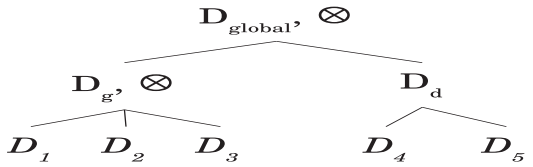

FIG. 1 – *Illustration d'un méta-modèle : il s'agit d'une hiérarchie des DMuS indiquant les dépendances.*

La Figure 1 illustre la notion de méta-modèle. Les nœuds marqués $\otimes$ expriment les contraintes de concordance, tandis que les autres expriment l'absence de contrainte entre les diagnostics des sous-branches. Dans cet exemple, les DMoS de D_1, D_2 et D_3 doivent fournir des diagnostics (élémentaires) qui concordent avec celui de D_g. Les DMoS de D_4 et D_5 ne sont pas contraints, mais le diagnostic (intermédiaire) du DMuS D_d est contraint à être concordant avec le diagnostic global obtenu par la fusion de son diagnostic avec celui du DMuS D_g.

[1] On notera que $\oplus$ dispose des bonnes propriétés (symétrie et associativité) pour généraliser facilement cette formule à la fusion de plus de deux diagnostics.

2.3 Construction d'un diagnostic global et adaptation

Dans cette section, on décrit l'utilisation du méta-modèle pour construire le diagnostic global et adaptater des DMoS.

2.3.1 Construction du diagnostic global

La hiérarchie du méta-modèle est utilisée pour organiser la construction du diagnostic global par l'application successive des diagnostiqueurs à partir des DMoS (les feuilles de l'arbre). Tout d'abord, les DMoS construisent leur diagnostic élémentaire à partir des observations. Ensuite, les DMuS construisent récursivement les diagnostics intermédiaires par la fusion des diagnostics définis par le méta-modèle jusqu'à obtenir le diagnostic global. Dans la théorie de DS, la fusion des diagnostics est symétrique et associative, l'ordre dans lequel sont exécutés les DMuS n'a donc pas d'importance.

2.3.2 Détection d'un besoin d'adaptation et adaptation

Lorsqu'un DMuS a construit son diagnostic, il vérifie si celui-ci satisfait les contraintes que le méta-modèle lui impose. Si le diagnostic résultant ne satisfait pas la contrainte du méta-modèle, c'est l'indication qu'il y a eu une défaillance dans le diagnostic. Le DMuS fait alors une **proposition d'adaptation** formée par les observations en cours d'analyse et le diagnostic du DMuS. Lorsqu'aucune contrainte n'est exprimée, un DMuS ne peut pas proposer d'adaptation. Par exemple, on s'attend à ce que des thermomètres identiques donnent la même température s'ils sont plongés dans la même eau (méta-modèle exprimant la concordance des diagnostiqueurs). Si l'un des thermomètres indique une température très différente des autres, c'est probablement qu'il fonctionne mal et qu'il est judicieux de proposer une réparation.

Dans notre système, l'adaptation ne vise pas, comme pour Chair et Varshney (1986), à ajuster l'influence relative des différentes sources d'information pour arriver à un meilleur résultat, ni à permettre d'accueillir dynamiquement de nouveaux DMoS comme le propose le système MidFusion de Alex et al. (2008). Les DMuS appliquent un schéma de fusion qu'il ne s'agit pas d'adapter. L'adaptation vise à modifier les fonctions de diagnostic utilisées par les DMoS en faisant judicieusement appel aux fonctions d'adaptation. Par conséquent, la proposition d'adaptation est directement envoyée aux DMoS situés aux feuilles du sous-arbre ayant pour racine le DMuS qui l'a proposée.

Plus précisément, dans le cas de la contrainte de concordance des diagnostics, si un diagnostic d'entrée d ne concorde pas avec le diagnostic de sortie alors les DMoS du sous-arbre de d reçoivent une proposition d'adaptation. Pour limiter le sur-apprentissage et les adaptations à partir de contre-exemples, les propositions d'adaptation sont faites uniquement lorsque la décision de diagnostic du DMuS est sûre, *i.e.* $\max_{A\in\Omega^2} Bel(A) > s_h$, et que la décision de diagnostic en entrée est peu sûre , *i.e.* $\max_{A\in\Omega^2} Bel(A) < s_b$.

Les propositions d'adaptation sont traitées par les DMoS une fois que tous les DMuS ont proposés des adaptations. Si un DMoS a reçu plusieurs propositions d'adaptation, il utilise uniquement celle dont la croyance est la plus forte pour adapter sa fonction de diagnostic.

3 Détection d'intrusion à partir de logs HTTP

L'approche de diagnostic multi-sources adaptatif est maintenant appliquée à la détection d'intrusion dans des serveurs Web. Afin d'éviter les utilisations malveillantes des serveurs Web, il faut développer des systèmes capables de détecter et d'identifier les intrusions (*Intrusion Detection System* ou IDS). Pour éviter les tâches fastidieuses de mise à jour de la base des modèles d'intrusion, le système doit pouvoir s'adapter de manière autonome pour intégrer dynamiquement la détection de nouvelles intrusions.

Nous nous intéressons aux IDS permettant la détection d'intrusion dans des serveurs Web à partir du journal des connexions HTTP (logs HTTP). Les logs HTTP constituent un flux de données structurées et riches. Un log contient des lignes décrivant les requêtes successives qui sont parvenues au serveur que l'on cherche à protéger des intrusions. Chaque ligne du log, illustrée par la Figure 2, est composée de plusieurs champs décrivant la requête et le comportement du serveur, par exemple l'IP du client, la date, l'URL de la requête ou le *status code*, statut attribué à la requête par le serveur (*e.g.* 404 : erreur dans l'URL de la requête, 200 : requête accomplie avec succès).

FIG. 2 – *Exemple d'une ligne de log de serveur Apache (format Combined).*

3.1 Décisions de diagnostic possibles et fusion des diagnostics

Dans le cas de la détection d'intrusion, l'espace de discernement (Ω) est composée de trois parties disjointes correspondant aux trois décisions de diagnostic possibles pour les lignes de log : intrusion (I), normal (N) ou inconnu (U). Le type $inconnu$ sert à la normalisation des masses et traduit la méconnaissance dans la théorie de DS. L'expression de la méconnaissance nous permet de prendre en compte les imprécisions des DMoS. Il est, par exemple, difficile de décider de manière certaine du caractère intrusif d'une requête à partir de la seule connaissance du *status code*.

Comme l'espace de discernement est une partition ($\Omega = \{I, N, U\}$), il est possible de simplifier les formules de combinaison de deux diagnostics :

$$\forall A \in \Omega,\ (m_1 \oplus m_2)(A) = \frac{m_1(A)m_2(A)}{1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)},$$

avec

$$\begin{aligned}\sum_{B \cap C = \emptyset} m_1(B)m_2(C) = {} & m_1(N)m_2(I) + m_1(N)m_2(U) + m_1(U)m_2(I) \\ & + m_1(U)m_2(N) + m_1(I)m_2(N) + m_1(I)m_2(U)\end{aligned}$$

Un diagnostiqueur a détecté une intrusion lorsque la croyance associée à la décision I est au-dessus d'un seuil s_i ($Bel(I) > s_i$).

3.2 DMoS et méta-modèle du diagnostic

Pour réaliser le diagnostic multi-sources des logs, on utilise deux types de DMoS :
– les DMoS qui utilisent la ligne de log courante,
– les DMoS qui utilisent la transaction dont fait partie la ligne courante.

Une transaction est l'ensemble des requêtes qu'un client (identifié par son IP) a transmis au serveur dans une fenêtre temporelle fixée à 10min. On fait l'hypothèse que *toutes les lignes de logs d'une transaction sont de même nature*, c'est-à-dire qu'un utilisateur normal, ne tentera pas d'intrusion (au pire il aura un mauvais usage du serveur conduisant à des erreurs sans mauvaises intentions), tandis que si un utilisateur effectue des requêtes jugées intrusives, alors on considérera l'ensemble de ses requêtes dans la fenêtre temporelle comme potentiellement intrusives.

Cette hypothèse permet de proposer un méta-modèle du diagnostic, illustré par la Figure 3, dans lequel il n'y a pas de contrainte entre les DMoS à partir de la ligne, ni entre les DMoS à partir de la transaction. Mais le diagnostic de la ligne obtenu par D_l et celui de la transaction obtenu par D_t doivent concorder avec le diagnostic global.

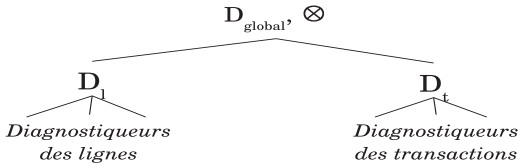

FIG. 3 – *Méta-modèle du diagnostic d'intrusion dans des logs HTTP.*

Nous avons implémenté des DMoS à partir des modèles d'intrusions et de requêtes normales suivants :
– distribution de caractères de l'URL d'une ligne de log,
– distribution de tokens[2] de l'URL d'une ligne de log,
– distribution Gaussienne de la proportion d'erreurs dans une transaction,
– distribution de caractères sur l'ensemble des URL d'une transaction.

Tous les modèles utilisés s'appuyant sur des distributions, leurs fonctionnements sont assez similaires. Par manque de place, on ne donne des détails que sur les fonctions du DMoS à partir de la distribution de caractères de l'URL. Pour ce diagnostiqueur, la fonction d'extraction d'information focalise l'attention uniquement sur l'URL de la ligne de log courante, et construit (*i.e.* structure l'information selon) une distribution normalisée des 256 caractères ASCII possibles d'une ligne courante. La fonction de diagnostic calcule la distance entre la distribution courante et un modèle de distribution appris à partir d'un ensemble d'intrusions connues. Un autre DMoS similaire utilise un ensemble de requêtes normales pour apprendre un modèle de requête normale. Moins la distance avec le modèle d'intrusion est importante, plus la masse m_I est grande. Lors de l'apprentissage préliminaire, la masse m_U est calculée par la proportion d'erreurs obtenue sur le jeu d'apprentissage. La fonction d'adaptation du diagnostiqueur ajoute la distribution de caractères de la ligne au modèle si la ligne correspond à une intrusion, mais ne fait rien si la ligne n'est pas une intrusion.

[2]Les tokens sont des unités sémantiques d'une URL distinguées par des séparateurs ('/', ' ?', '&').

4 Expérimentations et résultats

4.1 Données et implantation du système

L'objectif des évaluations du système est de montrer que la méthode d'adaptation en ligne adapte de manière pertinente les modèles d'intrusion. Les adaptations du système sont pertinentes si, d'une part, elles améliorent les performances du système n'utilisant pas les mécanismes d'adaptation (*i.e.* augmentation du rappel et de la précision), et si, d'autre part, elles permettent de découvrir les nouvelles intrusions, c'est-à-dire d'intégrer ces nouveaux exemples aux modèles d'intrusion initialement appris.

Nous avons recours à la simulation d'intrusion dans des logs réels. Nous disposons de logs d'accès aux serveurs Web des laboratoires INRIA/IRISA et INRIA/Sophia Antipolis chacun sur une période de 1 mois (respectivement, juin 2008 et mai 2007). Après un filtrage important, nous avons vérifié l'absence d'intrusion dans les données. Pour réaliser les expérimentations, nous utilisons une partie de ces logs dans laquelle nous introduisons artificiellement des transactions de requêtes intrusives tirées aléatoirement parmi 576 requêtes correspondant à des intrusions connues. Une portion de log d'une longueur fixée à 10000 lignes (environ 20 min de connexions) est extraite de l'ensemble des données originales et on y insère aléatoirement une grande quantité d'intrusions (environ 110 en moyenne, réparties dans une moyenne de 20 transactions).

Le rappel et la précision sont calculés pour plusieurs valeurs du seuil s_i (*cf.* Section 3.1) et on obtient ainsi des courbes rappel-précision.

Un sous-ensemble de 527 requêtes intrusives connues a été manuellement sélectionné pour servir d'ensemble d'apprentissage initial des modèles d'intrusions. Les 49 requêtes intrusives supprimées correspondent à un même type d'intrusion (requêtes permettant d'identifier des scripts se sachant vulnérables) dont aucune autre occurrence n'est présente dans l'ensemble d'apprentissage.

La méthode de détection d'intrusion a été implémentée sous la forme d'un logiciel d'analyse de logs de connexions Apache : *LogAnalyzer*[3]. Ce logiciel permet également la visualisation et l'exploration des logs (filtrage, analyse de la structure du serveur ou des transactions).

4.2 Expérimentation et résultats

Apport de la fusion des diagnostics multi-sources Nous mettons tout d'abord en évidence l'intérêt du diagnostic multi-sources et de la méthode de fusion des diagnostics en comparant les résultats obtenus à différents niveaux de la hiérarchie de notre méta-modèle de diagnostic (*cf.* Figure 3). La courbe (a) de la Figure 4 donne les courbes rappel/précision obtenues pour l'analyse d'un log par :

1. les DMoS seuls : distribution de tokens (`Tokens`) et distribution de caractères (`CD`),
2. les DMuS fusionnant les diagnostics obtenus respectivement à partir des lignes de logs (`LL`) et des transactions (`Trans`),
3. le DMuS global (`LL + Tr`).

[3] Plus d'informations sur le *LogAnalyzer* sont disponibles sur le site : *http ://www.irisa.fr/dream/LogAnalyzer/*.

Toutes les expérimentations menées avec différents jeux de données montrent un profil similaire des courbes. On observe systématiquement que les diagnostics obtenus par fusion ont un meilleur rappel et une meilleure précision que les diagnostics des DMoS :

- la courbe des lignes de logs (verte) a une AUC (aire sous la courbe) supérieure à celle des courbes des tokens (cyan) et des distributions de caractères (mauve),
- la courbe du diagnostic global (rouge) a une AUC supérieure à celui des transactions et des lignes.

Pertinence des adaptations On s'est ensuite intéressé à la pertinence des adaptations réalisées. On a pour cela effectué 26 expérimentations à partir de jeux simulés différents. En moyenne, 121.04 ($\pm$ 97.94) adaptations ont été réalisées. L'écart type élevé s'explique surtout par une grande disparité dans le nombre d'adaptations réalisées à partir de requêtes normales. On a constaté que les adaptations sont correctement réalisées à 95% ($\pm$0.02). C'est-à-dire qu'une adaptation dans le sens d'une ligne normale (resp. intrusive) n'a été lancée à partir d'une ligne intrusive (resp. normale) que dans 5% des adaptations. Ce résultat illustre la pertinence générale des choix d'adaptation. En pratique, cela signifie que le système limite l'intégration de requêtes intrusives dans le modèle normal et réciproquement.

De plus, dans 35% des expérimentations le système s'est adapté à partir d'une requête initialement inconnue. Ceci veut dire que dans plus d'un tiers des expérimentations, on a découvert le nouveau type d'intrusion et les modèles se sont bien enrichis pour les détecter ensuite.

Amélioration par l'adaptation Dans la mesure où les adaptations sont pertinentes, nous sommes en mesure d'attendre une amélioration des performances de la détection. Pour la mettre en évidence, on compare les résultats obtenus avec adaptation aux résultats obtenus sans adaptation pour des logs identiques.

La Figure 4 (b) donne les courbes rappel/précision typiques obtenues pour l'analyse d'un log. Les AUC des courbes obtenues avec une adaptation des modèles montrent des performances supérieures. La courbe orange est nettement au-dessus de la courbe cyan indiquant que l'adaptation de la distribution de tokens accroît efficacement les performances de ce DMoS. Ces améliorations accroissent les performances du DMuS des lignes : la courbe noire est également nettement au-dessus de la courbe verte. La fusion permet donc d'améliorer de façon significative les résultats obtenus à partir des indices pris individuellement. L'amélioration des performances globales du système (courbe jaune contre courbe rouge) n'est pas aussi importante que pour les autres courbes. D'une part, les performances de la détection à partir des transitions évoluent peu, et atténuent, par la fusion, les améliorations de la détection à partir des lignes. D'autre part, les performances globales sans adaptation sont initialement bonnes, l'amélioration qui peut être apportée par l'adaptation ne peut être très importante.

Le Tableau 1 présente des résultats obtenus à partir de 20 logs différents. Pour chaque expérimentation, on n'a conservé que les rappels et précisions correspondant à la F-Mesure maximale. Les résultats montrent que l'adaptation permet d'augmenter à la fois le rappel et la précision de la détection d'intrusion. En pratique,de nombreuses adaptations des modèles de requêtes normales sont réalisées (121 ($\pm$98) en moyenne). Ainsi, même si le nouveau type d'intrusion n'est pas systématiquement détecté, les performances globales du système sont tout de même meilleures dans plus de 60% des expérimentations et stables sinon.

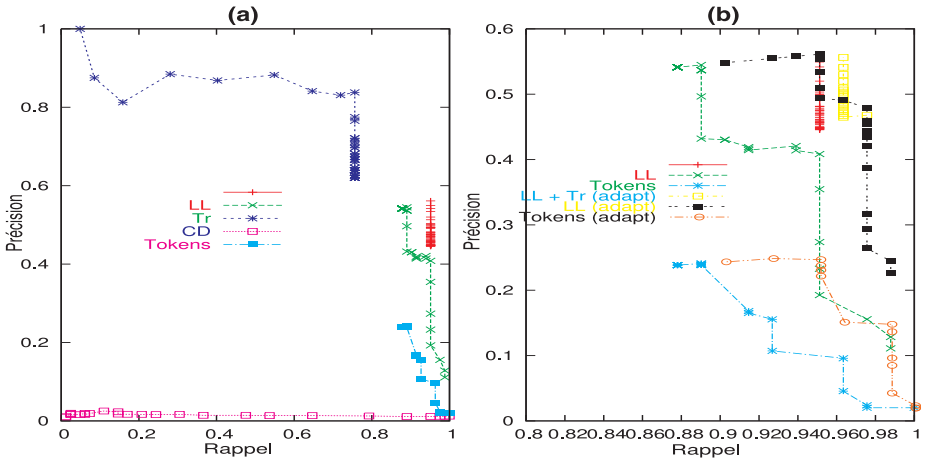

FIG. 4 – *Courbes rappel/précision pour la détection d'intrusion. (a) Comparaison des performances des DMoS et des DMuS, (b) Comparaison avec et sans adaptation.* `CD` *: Distribution de caractères,* `Tokens` *: Distribution des tokens,* `LL` *; Ligne de log,* `Tr` *: Transaction,* `(adapt)` *: avec adaptation.*

	Avec adaptation	Sans adaptation
Temps (s.)	12.68 ($\pm$ 0.46)	**11.83** ($\pm$ 0.38)
Rappel	**0.94** ($\pm$ 0.03)	0.93 ($\pm$ 0.04)
Précision	**0.81** ($\pm$ 0,25)	0.78 ($\pm$ 0.31)
F-mesure	**0.85** ($\pm$ 0.18)	0.81 ($\pm$ 0.22)

TAB. 1 – *Résultats des expérimentations comparant les détections d'intrusion avec et sans adaptation.*

Les temps de calcul sont très raisonnables puisque les 10000 lignes qui correspondent à environ 20 minutes d'enregistrement sur les serveurs testés sont traitées en quelques secondes. Cependant, nos expérimentations ont été volontairement menées avec peu d'indices, mais à terme, l'utilisation d'un grand nombre d'indices pourrait augmenter sensiblement le temps de traitement.

5 Autres approches de la détection d'intrusions

Les systèmes de détection d'intrusions à partir de logs HTTP utilisent pour la plupart des méthodes basées sur des signatures d'intrusions. Ces signatures sont comparées successivement et indépendamment sur chaque requête. Tombini et al. (2004) proposent comme modèle une liste de couples qui représentent la page demandée et les arguments éventuels des scripts.

Kruegel et Vigna (2003) ont introduit l'utilisation de modèles qui sont appris à partir d'une base d'exemples. Les modèles les plus simples reposent sur la distribution de caractères des URLs utilisées dans les requêtes et la distribution des tokens. Ingham et al. (2007) utilisent des automates déterministes finis pour représenter les URLs.

Mais pour Bass (2000), l'avenir des systèmes de détection d'intrusions passe par les méthodes de fusion d'information. Chaque signature est efficace pour détecter un certain type d'attaque, mais pas tous. Une approche qui permet de combiner différentes signatures apparaît comme une solution raisonnable pour détecter des intrusions de manière robuste et précise.

Les outils actuels de détection d'intrusions dans les serveurs Web (*e.g.* ModSecurity ou Snort de Roesch (1999)) nécessitent beaucoup de mises à jour manuelles des signatures d'intrusion. Des travaux cherchent donc à automatiser (totalement ou partiellement) la découverte de nouvelles intrusions, la construction de leurs signatures et la mise à jour des IDS. Kreibich et Crowcroft (2004) proposent *HoneyComb*, une méthode permettant d'automatiser la construction de signatures par l'observation d'un *HoneyPot*, *i.e.* un serveur Web peu protégé mais factice – personne n'est censé y accéder. Tout le trafic qui s'y passe peut être considéré comme malveillant et *HoneyComb* s'en sert pour automatiser la construction de nouvelles signatures. D'autres approches cherchent à adapter leurs modèles en ligne et parallèlement à la détection d'intrusion. Bojanic (2005) propose une méthode utilisant des HMM pour détecter des intrusions dans des séquences de commande système et Srinoy (2006) utilise des SVMs pour modéliser les intrusions et une technique d'intelligence collective (Swarm intelligence) pour permettre une adaptation dynamique des modèles d'intrusion.

6 Conclusion

Nous avons proposé une méthode de diagnostic multi-sources adaptatif qui s'inspire de la fusion d'information pour combiner les diagnostics de DMoS. Un méta-modèle du diagnostic est construit à partir de contraintes connues entre les DMoS. La comparaison entre les diagnostics réalisés et le méta-modèle permet de proposer les adaptations qu'effectuent individuellement les DMoS sur eux-mêmes.

Nous avons appliqué cette méthode à la conception d'un système adaptatif de détection d'intrusion à partir de logs Apache et nous avons montré que 1) les choix d'adaptation étaient pertinents, 2) que le système découvre de nouvelles intrusions et 3) que les performances de détection d'intrusion étaient améliorées.

Actuellement, la fusion centralise toutes les adaptations et les appliquent *a posteriori* de la construction du diagnostic. La première perspective de ce travail est de proposer une approche multi-agents du diagnostic adaptatif multi-sources dans laquelle les décisions d'adaptation seraient discutées entre agents conjointement au calcul du diagnostic. Le seconde perspective est de permettre l'expression dans le méta-modèle de nouveaux types de contraintes entre les diagnostiqueurs.

Références

Alex, H., M. Kumar, et B. Shirazi (2008). MidFusion : An adaptive middleware for information fusion in sensor network applications. *Information Fusion 9*(3), 332–343.

Bass, T. (2000). Intrusion detection systems and multisensor data fusion. *Communication of the ACM 43*(4), 99–105.

Bojanic, I. (2005). On-line adaptive IDS scheme for detecting unknown network attacks using HMM models. Master's thesis, University of Maryland.

Chair, Z. et P. K. Varshney (1986). Optimal data fusion in multiple sensor detection system. *IEEE Transactions on Aerospace and Electronic Systems 22*(1), 98–101.

De Kleer, J. (2007). Dynamic domain abstraction through meta-diagnosis. In I. Miguel et W. Ruml (Eds.), *Proceedings of the 7th International Symposium on Abstraction, Reformulation, and Approximation (SARA)*, pp. 109–123.

Fromont, É., R. Quiniou, et M.-O. Cordier (2005). Learning rules from multisource data for cardiac monitoring. In *Proceedings of the 10th conference on Artificial Intelligence in Medicine (AIME'05)*, pp. 484–493.

Ingham, K. L., A. Somayaji, J. Burge, et S. Forrest (2007). Learning DFA representations of HTTP for protecting web applications. *Computer Networks 51*, 1239–1255.

Kreibich, C. et J. Crowcroft (2004). HoneyComb : creating intrusion detection signatures using honeypots. *SIGCOMM Computer Communication Review 34*(1), 51–56.

Kruegel, C. et G. Vigna (2003). Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security (CCS '03)*, pp. 251–261.

Roesch, M. (1999). Snort : Lightweight intrusion detection for networks. In *Proceedings of the 13th Conference on Systems Administration*, pp. 229–238.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

Srinoy, S. (2006). An adaptive IDS model based on swarm intelligence and support vector machine. In *Proceedings of the International Symposium on Communications and Information Technologies (ISCIT '06)*, pp. 584–589.

Tombini, E., H. Debar, L. Mé, et M. Ducassé (2004). A serial combination of anomaly and misuse IDS applied to HTTP traffic. In *Proceedings of the 20th annual computer security application conference (ACSAC '04)*, pp. 428–437.

Summary

An adaptive monitoring system aims at monitor and diagnose a system while adapting to its evolution. This requires the adaptation of the diagnosers which specify or enrich their own model to evolve with the system at the same time. We propose a multi-source diagnosis framework based on the information fusion. Relations between mono-source diagnosers are provided by the designer and define a meta-model of diagnosis. The compatibility of the current diagnosis with the meta-model is checked online. When one relationship is not satisfied, the underlying mono-source diagnosers are adapted.

We apply this approach to the design of an adaptive intrusion detection system for a HTTP connection log stream from a Web server. The evaluations show that the system improves its ability to detect known intrusions and that it discovers new kinds of attack.