République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Ibn Khaldoun– Tiaret

Faculté des Mathématiques et de l'Informatique

Département Informatique

MEMOIRE EN VUE DE L'OBTENTION DU DIPLOME DE MAGISTER

SPECIALITE : Informatique

OPTION : Informatique **R**épartie et **M**obile **(IRM)**

SUJET DU MEMOIRE

Une Approche de Coopération Entre IDS Basée sur SOA

Réalisé par :

Khadidja BEKKI

Soutenu le 09/04/2016 devant le jury composé de :

Mr. Amar BALLA	Professeur (ESI) - ALGER	Président
Mr. Omar NOUALI	Directeur de recherche (CERIST) - ALGER	Examinateur
Mr. Rachid CHALAL	Maitre de conférences A (ESI) - ALGER	Examinateur
Mr. Youcef DAHMANI	Maitre de conférences A (UIK) – TIARET-	Directeur du mémoire

ANNEE UNIVERSITAIRE 2014/2015

Résumé

Avec la prolifération du réseau Internet et l'émergence de menaces protéiformes, la sécurisation des infrastructures informatiques s'impose comme un enjeu stratégique majeur. Divers mécanismes de protection ont été développés, parmi lesquels les systèmes de détection d'intrusion (IDS) constituent une pierre angulaire. L'implémentation des IDS s'est rapidement généralisée au sein des environnements organisationnels et des systèmes critiques. Ainsi, plusieurs modèles d'IDS ont vu le jour, chacun présentant des atouts et des limites spécifiques. Pour pallier ces insuffisances, l'intégration de multiples IDS s'est avérée nécessaire afin d'exploiter leur complémentarité et de renforcer leur efficacité. Cependant, cette interconnexion se heurte à une hétérogénéité structurelle et fonctionnelle des IDS, qui adoptent des protocoles de communication, des formats de données et des paradigmes conceptuels distincts. De plus, la coopération inter-IDS doit être suffisamment modulable pour permettre aux responsables de la cybersécurité de l'adapter dynamiquement en fonction des exigences sécuritaires et d'optimiser les mécanismes de réponse en temps opportun. Notre travail vise à proposer une approche de coopération adaptable permettant de transcender ces disparités et d'améliorer la réactivité des IDS. À cet effet, l'architecture orientée services (SOA) constitue une solution pertinente pour garantir l'interopérabilité et la gestion de l'hétérogénéité des systèmes. Par ailleurs, l'adoption des règles ECA (Event-Condition-Action), intrinsèquement réactives, optimise la souplesse du processus collaboratif. Dans ce cadre, nous avons élaboré un scénario de coopération flexible entre deux IDS hétérogènes en nous appuyant sur l'architecture SOA.

Mots-clés : Cybersécurité, systèmes de détection d'intrusion, interopérabilité, SOA, adaptation dynamique.

Abstract

With the proliferation of the Internet and the emergence of multifaceted cyber threats, securing IT infrastructures has become a critical strategic challenge. Various protection mechanisms have been developed, among which Intrusion Detection Systems (IDS) constitute a cornerstone. The deployment of IDS has rapidly expanded within organizational environments and critical systems. Consequently, multiple IDS models have been designed, each offering specific advantages and limitations. To mitigate these shortcomings, integrating multiple IDS has proven necessary to leverage their complementarities and enhance their effectiveness. However, this interconnection faces structural and functional heterogeneity among IDS, which rely on different communication protocols, data formats, and conceptual paradigms. Moreover, inter-IDS cooperation must be sufficiently adaptable, allowing cybersecurity managers to dynamically adjust security strategies and optimize response mechanisms in real-time. This research aims to propose a flexible cooperation approach that overcomes these disparities and enhances IDS responsiveness. In this context, Service-Oriented Architecture (SOA) presents a viable solution for ensuring interoperability and managing system heterogeneity. Additionally, the adoption of Event-Condition-Action (ECA) rules, which are inherently reactive, improves the adaptability of the cooperative process. Based on this framework, we have developed a flexible cooperation scenario between two heterogeneous IDS using the SOA architecture.

Keywords: Cybersecurity, Intrusion Detection Systems, interoperability, SOA, dynamic adaptation.

TABLE DES MATIERES

Table des Matières

TABLE DES TABLEAUX

TABLE DES FIGURES

Table des Figures

Introduction Générale

Introduction générale

Dans un monde où les réseaux et les infrastructures informatiques jouent un rôle prépondérant, ces derniers constituent désormais une composante essentielle des administrations publiques, des activités économiques et du quotidien des citoyens. Cependant, cette omniprésence s'accompagne d'une augmentation exponentielle des menaces cybernétiques, alimentée par la prolifération d'outils de piratage sophistiqués, accessibles librement en ligne. De fait, l'exécution d'attaques informatiques ne requiert plus une expertise technique avancée, rendant ainsi possible l'orchestration d'attaques dévastatrices par des individus aux compétences limitées. Par ailleurs, l'interconnexion massive des systèmes numériques accentue l'exposition aux vulnérabilités, engendrant une recrudescence des risques pesant sur la sécurité des infrastructures informatiques. Dès lors, la nécessité d'adopter des stratégies de défense robustes devient impérative, tant pour les particuliers que pour les entreprises et les institutions étatiques.

Parmi les mécanismes fondamentaux de protection, les systèmes de détection d'intrusion (IDS) occupent une place centrale. Ces dispositifs sont conçus pour identifier les comportements malveillants menaçant l'intégrité des systèmes informatiques et générer des alertes destinées aux responsables de la sécurité, leur permettant ainsi d'anticiper et de prévenir les risques d'intrusion. Toutefois, en raison de la diversité et de la complexité croissantes des cyberattaques, aucun IDS ne peut, à lui seul, garantir une détection exhaustive et efficiente. Chaque solution présente des forces et des limitations inhérentes à ses techniques d'analyse et aux paradigmes sur lesquels elle repose. Par conséquent, il devient impératif d'instaurer une coopération entre plusieurs IDS afin d'exploiter leur complémentarité et d'atténuer leurs insuffisances respectives.

Diverses approches ont été explorées pour améliorer la collaboration entre différentes stratégies de détection et accroître l'efficacité des IDS. Cependant, l'objectif d'une détection optimale ne peut être pleinement atteint si cette coopération ne surmonte pas l'hétérogénéité des IDS, ne permet pas une reconfiguration dynamique en fonction de l'évolution des menaces, et ne confère pas une flexibilité accrue aux administrateurs de sécurité.

Dans cette optique, nous nous intéressons particulièrement à l'architecture orientée services (SOA), qui constitue une solution pertinente pour assurer l'interopérabilité et la gestion de l'hétérogénéité des systèmes. Notre problématique porte spécifiquement sur l'amélioration de la flexibilité dans la coopération inter-IDS. Cette flexibilité se définit comme la capacité d'un système à s'adapter et à réagir efficacement aux évolutions et aux exigences contextuelles. Elle représente un enjeu stratégique majeur pour les administrateurs de sécurité, soucieux d'accroître leur agilité et leur réactivité afin d'anticiper et de neutraliser les menaces émergentes.

Dans ce cadre, nous proposons CIIDS-SOA (*Coopération Inter IDS basée sur SOA*), un modèle de coopération conçu pour établir une interaction dynamique et adaptable entre des IDS hétérogènes.

Ce mémoire s'articule autour de quatre chapitres :

- Les deux premiers chapitres sont consacrés à l'état de l'art :

- Le premier chapitre introduit les concepts d'intrusion informatique et les principes des systèmes de détection d'intrusions.

- Le deuxième chapitre explore l'architecture orientée services, en détaillant la technologie des services web et ses différentes méthodes de composition.

- Le troisième chapitre examine l'application de SOA dans le domaine de la détection d'intrusions, en mettant en lumière des travaux antérieurs pertinents.

- Le dernier chapitre expose notre contribution visant à instaurer une coopération flexible entre IDS hétérogènes et décrit l'implémentation de l'approche proposée.

Enfin, ce mémoire se conclut par une synthèse générale et une discussion sur les perspectives d'amélioration et d'extension de notre travail.

Chapitre -1-

Les Systèmes de Détection d'Intrusions

1. Introduction

À l'ère de la transformation numérique, les réseaux informatiques constituent une infrastructure essentielle, aussi bien pour les particuliers que pour les entreprises et les institutions étatiques. Ils facilitent une diversité de services tels que le partage de données et de ressources, la messagerie électronique ainsi que le commerce électronique. Cette généralisation de l'usage des réseaux a conduit à une augmentation exponentielle du nombre d'utilisateurs cherchant à exploiter leurs potentialités. Toutefois, cette expansion s'accompagne inévitablement d'une recrudescence des menaces informatiques et des intrusions malveillantes, rendant impérative l'instauration de stratégies de protection avancées. La sécurisation des systèmes d'information s'impose dès lors comme un enjeu majeur, nécessitant la mise en place de dispositifs adaptés visant à préserver la confidentialité, l'intégrité et la disponibilité des données sensibles.

Parmi les mécanismes de défense les plus incontournables figure le système de détection d'intrusions (*Intrusion Detection System - IDS*). Ce dispositif a pour vocation d'identifier les comportements suspects ou anormaux affectant un système informatique et de générer des alertes permettant aux administrateurs de sécurité d'adopter des mesures préventives et correctives.

Ce chapitre s'articule autour des aspects fondamentaux de la sécurité informatique. Nous débuterons par une exploration des concepts de base en matière de cybersécurité avant d'analyser les principales typologies d'attaques. Par la suite, nous approfondirons le domaine de la détection d'intrusions en détaillant les caractéristiques essentielles des IDS, en établissant une classification rigoureuse et en présentant quelques solutions existantes.

2. Notions de sécurité

Les États, les infrastructures critiques, les organisations ainsi que les individus sont exposés à des cybermenaces de plus en plus sophistiquées, mettant en péril l'intégrité des systèmes et la confidentialité des données. L'impact des attaques varie en fonction de la criticité des informations traitées, faisant de la sécurité informatique un impératif stratégique. Dans ce contexte, la sécurité des systèmes d'information repose sur trois propriétés fondamentales [1] :

2.1. Propriétés de sécurité

– Confidentialité : Assure que l'accès aux données est strictement réservé aux entités autorisées, empêchant toute divulgation ou consultation non légitime.

– Intégrité : Garantit que les informations stockées ou échangées ne subissent aucune altération non autorisée, assurant ainsi leur exactitude et leur fiabilité.

– Disponibilité : Veille à ce que les ressources et les services soient accessibles en permanence aux utilisateurs légitimes, minimisant ainsi les interruptions susceptibles de nuire aux activités opérationnelles.

2.2. Politique de sécurité

Une politique de sécurité constitue un cadre stratégique visant à préserver ces trois propriétés. Elle repose sur des principes directeurs, des mesures préventives et des mécanismes de contrôle permettant de minimiser les risques et de garantir une résilience efficace du système d'information.

Les principaux axes d'une politique de sécurité incluent :

- Gestion des accès : Implémentation de stratégies de classification et de restriction des privilèges en fonction des rôles et responsabilités des utilisateurs.
- Mécanismes d'authentification et de contrôle : Déploiement de protocoles robustes (authentification forte, gestion des identités, logs d'audit) pour limiter les accès non autorisés.
- Plan de continuité et de reprise d'activité : Mise en place de solutions permettant d'assurer la disponibilité des services en cas de panne ou d'attaque.

Une politique efficace doit également anticiper les scénarios de vulnérabilités et définir des protocoles de réponse adaptés, garantissant une adaptation dynamique aux menaces émergentes [2].

2.3. Différents aspects de la sécurité

L'implémentation d'une politique de sécurité doit être envisagée à plusieurs niveaux:

- Niveau logiciel : Application de mécanismes de protection tels que l'authentification multi-facteurs, le chiffrement des communications et la gestion rigoureuse des accès.

- Niveau physique : Sécurisation des infrastructures matérielles contre les menaces physiques (sabotage, vol, incendies, catastrophes naturelles, etc.), par le biais de dispositifs de contrôle d'accès et de surveillance.

- Niveau humain : Sensibilisation et formation des utilisateurs aux bonnes pratiques en matière de cybersécurité, car la négligence humaine demeure un vecteur d'attaque majeur.

- Niveau technologique : Mise à jour régulière des logiciels, veille technologique et renforcement des protocoles de sécurité pour pallier les nouvelles vulnérabilités et contrer les attaques émergentes.

2.4. Mécanismes de sécurité

L'application rigoureuse d'une politique de sécurité repose sur un ensemble de mécanismes techniques et organisationnels permettant d'assurer une protection efficace contre les cybermenaces [2] :

- **Authentification** : Vérification stricte de l'identité des utilisateurs à travers des systèmes robustes (biométrie, certificats numériques, authentification forte, etc.).

- **Contrôle d'accès** : Définition de privilèges et de droits spécifiques accordés aux utilisateurs en fonction de leurs attributions, réduisant ainsi les risques de compromission.

- **Pare-feux** : Filtrage et régulation du trafic réseau pour empêcher les accès non autorisés et bloquer les tentatives d'intrusion.

- **Systèmes de détection d'intrusions (IDS)** : Surveillance active des événements réseau et détection des activités suspectes ou malveillantes.

- **Chiffrement des données** : Protection des informations sensibles par l'application d'algorithmes cryptographiques garantissant la confidentialité et l'intégrité des échanges.

- **Audit et journalisation** : Enregistrement systématique des événements critiques afin de faciliter l'analyse des incidents et d'améliorer les stratégies de réponse aux attaques.

Face à l'essor des cybermenaces et à la complexité croissante des attaques, la mise en place de stratégies de sécurité rigoureuses et évolutives est devenue une nécessité absolue. La sécurité informatique repose sur des principes fondamentaux, articulés autour de la confidentialité, de l'intégrité et de la disponibilité des données. Afin de préserver ces propriétés, il est indispensable d'adopter une politique de sécurité robuste, intégrant des mécanismes techniques et organisationnels sophistiqués.Dans le cadre de cette étude, nous

approfondirons l'un des mécanismes de défense les plus stratégiques : le système de détection d'intrusions (IDS). Ce dernier constitue un outil indispensable pour identifier et contrer les menaces ciblant les infrastructures numériques.

3. Attaques et Intrusions Informatiques

Les cyberattaques représentent une menace grandissante pour l'intégrité des infrastructures numériques, affectant aussi bien les entreprises que les institutions étatiques et les particuliers. Selon [3], une attaque informatique se définit comme l'exécution d'actions malveillantes exploitant une vulnérabilité identifiée dans un système.

Une vulnérabilité correspond à une faille, qu'elle soit d'origine matérielle, logicielle ou humaine, permettant à un acteur malveillant de contourner. Les cybercriminels s'appuient sur ces failles pour infiltrer les systèmes, en obtenir le contrôle ou exfiltrer des informations sensibles de manière clandestine.

Une intrusion résulte d'une attaque partiellement ou totalement réussie, impliquant un accès non autorisé à un réseau ou à un système. Ces intrusions peuvent être d'origine externe (attaquants distants) ou interne (personnel, clients, partenaires commerciaux), ce qui souligne la nécessité de mettre en place des mesures de détection avancées.

3.1. Taxonomie des Attaques

Les motivations des attaquants varient selon la nature des cibles et les objectifs poursuivis. Les chercheurs ont proposé différentes classifications des attaques informatiques basées sur des critères distincts.

3.1.1. Classification selon le Type d'Attaque

Cette classification [4] regroupe les attaques en quatre catégories majeures :

- **Scan (Probing)** : Cette technique consiste à envoyer divers paquets réseau à un hôte ou un réseau afin d'identifier les services actifs et leurs éventuelles vulnérabilités [5]. L'attaquant obtient ainsi des informations précieuses sur la structure du réseau, les systèmes d'exploitation déployés et les applications en cours d'exécution.

- **Attaques par déni de service (DoS)** : Ces attaques visent à saturer les ressources d'un système en exploitant des failles logicielles ou en générant un volume excessif de requêtes. On distingue deux types principaux [5] :

- **Exploitation des vulnérabilités** : Comme l'attaque Ping of Death, qui envoie des paquets

trop volumineux pour être traités par la cible.

• **Inondations (Flooding)** : Comme l'attaque SYN Flood, qui surcharge le serveur en multipliant les demandes de connexion incomplètes.

• **Compromission des systèmes** : Ces attaques exploitent des vulnérabilités connues pour obtenir un accès privilégié aux hôtes cibles [4].

• **Maliciels (virus, vers, chevaux de Troie)** : Ces programmes malveillants se propagent au sein d'un réseau et compromettent les machines infectées [4].

3.1.2. Classification selon la Source de l'Attaque

Selon [4], les attaques peuvent être initiées depuis :

• Une seule source : L'attaque est orchestrée à partir d'un unique point d'origine.

• Plusieurs sources : Les attaques distribuées (DDoS) impliquent plusieurs machines réparties sur divers réseaux, rendant leur détection plus complexe.

3.1.3. Autres Classifications

• **Attaques passives et actives** :

– *Attaques passives* : L'attaquant observe le trafic sans modifier les ressources du système.

– *Attaques actives* : Elles visent à altérer les ressources ou à en prendre le contrôle [6].

• **Attaques internes et externes** :

– *Externes* : Effectuées par des utilisateurs non autorisés tentant d'accéder illégalement aux systèmes.

– *Internes* : Provoquées par des utilisateurs légitimes cherchant à outrepasser leurs privilèges ou à exploiter des failles internes [7].

3.2. Exemples d'Attaques

Les cyberattaques évoluent continuellement, exploitant des failles variées. Parmi les plus courantes [6] :

• **IP Spoofing** : Falsification de l'adresse IP source pour contourner les mécanismes d'authentification.

• **Sniffing** : Interception du trafic réseau pour capturer des informations sensibles.

• **SYN Flood** : Saturation des connexions TCP via l'envoi massif de requêtes SYN sans

finalisation du handshake.

- **Logiciels malveillants** : Programmes destructeurs visant la compromission des systèmes, Ils consistent à exécuter desactionsmalveillantestelles que la perturbation d'une activité donnée ou la collecte des informationsconfidentielles [8], les virus, les vers et cheval de Troie sont des types de Logiciels malveillants.

- **Ingénierie sociale** : Manipulation psychologique visant à obtenir des informations confidentielles auprès d'utilisateurs crédules.

- **Brute Force & Dictionnaire** : Essai systématique de mots de passe pour accéder à des comptes protégés.

- **Attaques électroniques (Tempest)** : Espionnage des émissions électromagnétiques pour extraire des données sensibles [9].

4. Contremesures

Face à la prolifération des cybermenaces, plusieurs mécanismes de défense ont été développés afin d'assurer la protection des infrastructures numériques.

4.1. Pare-feu

Un pare-feu filtre le trafic réseau en appliquant des règles de sécurité restrictives. Il empêche les accès non autorisés, bien que son efficacité soit limitée face aux menaces internes [17].

4.2. Système de Détection d'Intrusions (IDS)

Un IDS analyse les activités d'un réseau ou d'un système pour identifier les comportements suspects. Nous détaillerons son fonctionnement dans la section 5.

4.3. Systèmes de Leurres (Honeypots)

Les honeypots sont des systèmes factices destinés à piéger les attaquants et à analyser leurs méthodes d'attaque.

4.4. Antivirus

Les antivirus détectent et éliminent les logiciels malveillants. Leur efficacité repose sur des bases de signatures mises à jour régulièrement.

5. Détection d'Intrusions

5.1. Définition et Fonctionnement

La détection d'intrusions consiste à surveiller et analyser les événements d'un système afin d'identifier des tentatives de compromission [11]. Un IDS automatise cette surveillance et génère des alertes lorsqu'une anomalie est détectée [12].

5.2. Évaluation de l'Efficacité d'un IDS

L'efficacité d'un IDS est mesurée selon plusieurs critères [14,15] :

- Exactitude : Capacité à identifier correctement les intrusions tout en minimisant les fausses alertes.
- Performance : Rapidité de traitement des événements d'audit.
- Résilience : Capacité à maintenir son fonctionnement malgré des attaques ciblées.
- Opportunité : Détection et signalement rapides permettant une réaction en temps réel.

5.3. Classification des IDS

Depuis les travaux d'Anderson [5] et de Ning [16], diverses classifications des IDS ont été proposées. La figure 1 illustre ces principaux critères de classification.

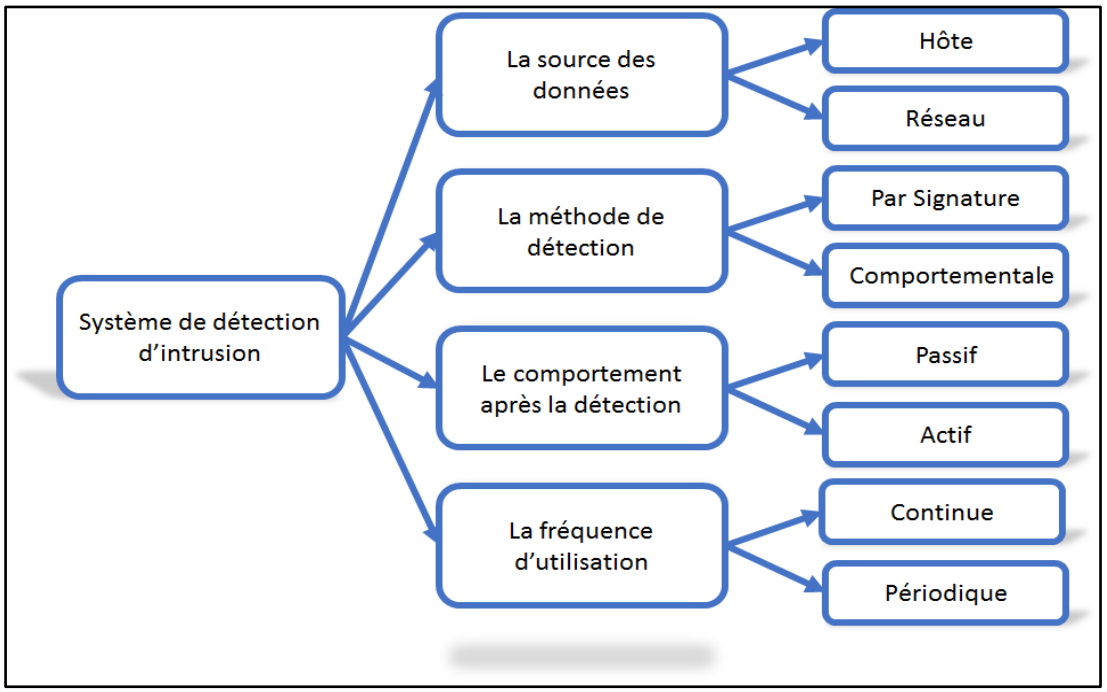

Figure1: Classification des Systèmes de Détection d'Intrusions (IDS) selon Divers Critères

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies.

5.3.1. Sources de Données

a. Systèmes de Détection d'Intrusion Basés sur l'Hôte (HIDS)

Les HIDS (*Host-Based Intrusion DetectionSystems*) surveillent un hôte spécifique en analysant ses journaux système, ses processus actifs, l'activité de ses applications et toute modification affectant ses fichiers critiques ou ses paramètres de configuration [11]. Ils permettent d'identifier des anomalies localisées et sont généralement installés sur des serveurs stratégiques hébergeant des données sensibles.

➤ ***Avantages***

- Haute précision analytique, capable d'identifier des modifications subtiles affectant le système surveillé.
- Capacité de fonctionnement en environnements chiffrés, contrairement aux IDS basés sur le réseau.

➤ ***Inconvénients***

- Consommation de ressources importante pouvant dégrader les performances de l'hôte.
- Vulnérabilité à des attaques réussies pouvant altérer ou masquer les journaux d'événements.
- Complexité de gestion, chaque HIDS nécessitant une configuration indépendante.

b. Systèmes de Détection d'Intrusion Basés sur le Réseau (NIDS)

Face à la recrudescence des attaques ciblant les infrastructures réseau (*port scanning*, *spoofing*, *flooding*), les NIDS (*Network-Based Intrusion DetectionSystems*) ont été développés pour surveiller et analyser le trafic réseau en interceptant les paquets et en examinant leurs contenus [4, 18].

➤ ***Avantages***

- Surveillance étendue sur l'ensemble du réseau, minimisant les coûts de déploiement.
- Moins vulnérables aux attaques ciblant des hôtes individuels, car indépendants des machines analysées.

➤ ***Inconvénients***

- Difficulté d'analyse des communications chiffrées.

- Sensibilité aux environnements réseau à haut débit, où des pertes de paquets peuvent altérer leur efficacité.

c. Audits Applicatifs

Certains IDS sont spécialement conçus pour surveiller les applications en analysant les journaux qu'elles génèrent. Ils constituent une sous-catégorie des HIDS, mais avec une focalisation sur la couche applicative.

d. IDS Hybrides

Les IDS hybrides intègrent les fonctionnalités des HIDS et des NIDS, offrant une visibilité accrue sur l'ensemble du réseau et des hôtes. Leurs sondes peuvent fonctionner en tant que NIDS ou HIDS selon leur positionnement et envoient des alertes consolidées à un serveur central chargé de corréler les informations.

5.3.2. Méthodes d'Analyse

Les IDS peuvent être classés en fonction de la méthode utilisée pour détecter une intrusion. Deux approches principales existent :

5.3.2.1. Approche par Signatures (Détection Basée sur des Scénarios)

Les IDS à détection par signature s'appuient sur une base de données contenant des empreintes numériques d'attaques connues. Lorsqu'un comportement suspect correspond à l'une de ces signatures, une alerte est déclenchée [11, 14, 19].

Techniques Employées

- Reconnaissance de formes [20, 21] : Assimile le fichier d'audit du système à une séquence d'événements et recherche la présence de sous-séquences caractéristiques d'attaques.
- Systèmes experts [14] : Exploitent des bases de règles pour modéliser des scénarios d'attaques et établir des diagnostics via un moteur d'inférence.

➤ Avantages

- Grande fiabilité dans la détection d'attaques connues.
- Réduction du taux de fausses alertes grâce à l'identification précise des menaces.

➤ Inconvénients

- Incapacité à identifier des attaques inédites.

- Nécessite une mise à jour constante pour inclure de nouvelles signatures.

5.3.2.2. Approche Comportementale (Détection d'Anomalies)

L'approche comportementale repose sur l'observation et l'apprentissage des habitudes d'un système ou d'un utilisateur. Un modèle de référence est établi, puis toute déviation statistiquement significative est considérée comme une potentielle intrusion [19].

Méthodes de Modélisation

- Modèle statistique : Évaluation des variations des paramètres système (ex. : charge CPU, fréquences d'accès, durée des sessions) pour détecter des anomalies.
- Réseaux de neurones : Exploitation des capacités d'apprentissage automatique pour identifier des comportements atypiques et anticiper les menaces émergentes.

➤ Avantages

- Identification possible d'attaques inédites.
- Adaptabilité aux évolutions du système surveillé.

➤ Inconvénients

- Risque élevé de faux positifs, particulièrement dans des environnements dynamiques.
- Vulnérabilité aux attaques furtives introduisant progressivement des comportements malveillants dans le profil de référence.

5.3.2.3. Comparaison entre l'Approche Comportementale et l'Approche par Scénarios

Tableau 01 : Comparaison entre l'Approche par Signature et l'Approche Comportementale des IDS

Critère	Approche par Signature	Approche Comportementale
Détection d'attaques inconnues	Non	Oui
Taux de faux positifs	Faible	Élevé
Maintenance	Mise à jour fréquente requise	Apprentissage dynamique
Compréhension des alertes	Alerte explicite (scénario connu)	Difficulté d'interprétation (anomalie statistique)

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies.

5.3.3. Fréquence d'Analyse

Les IDS peuvent analyser les événements en temps réel ou à intervalle régulier [4] :

- Analyse en continu (temps réel) : Surveillance instantanée, permettant des réactions immédiates en cas d'intrusion.
- Analyse périodique : Traitement différé des données d'audit, optimisant les ressources système mais retardant la détection des menaces.

5.3.4. Comportement après Détection

Un IDS peut réagir de manière active ou passive en cas de menace détectée.

- Réponses actives : Déclenchement automatique de contre-mesures telles que le blocage d'une adresse IP suspecte ou la reconfiguration de pare-feu et routeurs.
- Réponses passives : Signalement de l'intrusion via des alertes ou des logs, laissant l'initiative de la réaction aux administrateurs de sécurité.

5.4. Architecture des Systèmes de Détection d'Intrusions

Deux architectures principales sont utilisées pour l'implémentation des IDS [4,10] :

- Architecture centralisée : Un seul système collecte et analyse toutes les données. Bien que simple à gérer, cette approche peut être inefficace face aux attaques distribuées.
- Architecture distribuée : Plusieurs analyseurs répartis sur différents segments du réseau collaborent pour détecter les intrusions. Cette architecture est particulièrement adaptée aux attaques DDoS et aux menaces persistantes avancées.

6. Placement des Systèmes de Détection d'Intrusions (IDS)

Le positionnement des systèmes de détection d'intrusions (IDS) dans une infrastructure réseau est un élément fondamental influant sur leur efficacité. Selon l'emplacement choisi, l'IDS peut offrir une visibilité accrue sur certaines attaques et permettre une meilleure corrélation des événements de sécurité.

6.1. Positionnement des NIDS

La figure2 illustre les principaux emplacements stratégiques des IDS réseau (*Network-Based Intrusion DetectionSystems – NIDS*).

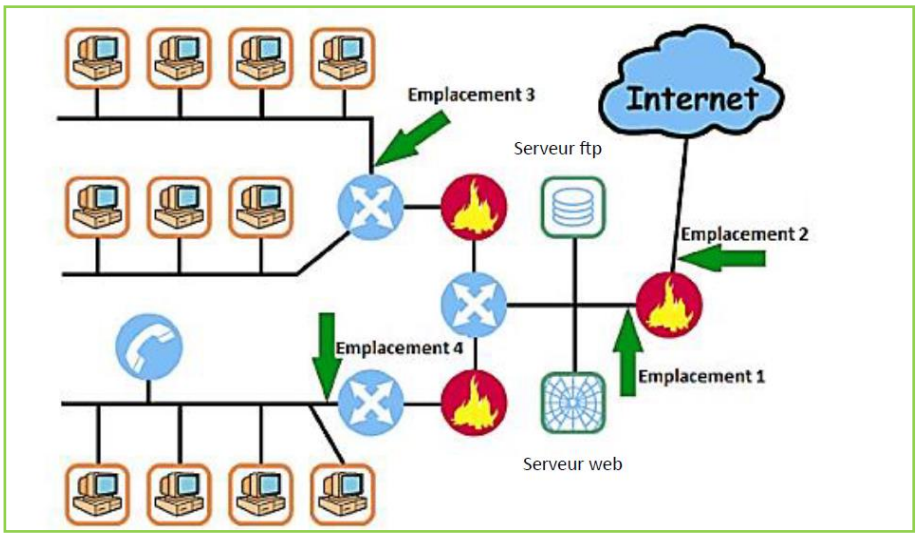

Figure 2 : Emplacement stratégique des IDS dans un réseau informatique [12].

- Emplacement 1 **:** Déploiement à l'entrée des segments critiques du réseau (ex. : serveurs sensibles, bases de données confidentielles) pour identifier les tentatives d'intrusion ciblant ces ressources stratégiques.
- Emplacement 2 : Positionnement en amont du pare-feu afin de superviser l'intégralité du trafic entrant et d'évaluer la pertinence des règles de filtrage appliquées.
- Emplacement 3 : Surveillance des segments internes du réseau pour détecter des activités suspectes post-intrusion, révélant des mouvements latéraux ou des compromissions persistantes.
- Emplacement 4 : Intégration sur des réseaux à haute criticité pour assurer une protection proactive contre les attaques sophistiquées.

6.2. Positionnement des HIDS

Les systèmes de détection d'intrusions basés sur l'hôte (HIDS) viennent en complément des NIDS en offrant une visibilité accrue sur l'activité locale des machines surveillées. Leur installation est prioritairement recommandée sur :

- Les serveurs stratégiques hébergeant des données sensibles ou des services critiques.
- Les postes administrateurs susceptibles d'être ciblés par des attaques visant à obtenir des privilèges élevés.
- Les machines ayant accès à des ressources sensibles, afin de superviser les activités suspectes des utilisateurs légitimes.

7. Limites des Systèmes de Détection d'Intrusions (IDS)

Bien qu'essentiels pour la sécurisation des infrastructures numériques, les IDS présentent certaines limites inhérentes à leur conception et à leur mode d'exploitation [13] :

- Absence de capacité préventive : Les IDS se limitent à la détection et à l'alerte, sans mécanismes intégrés de mitigation active.
- Intervention humaine requise : Leur gestion et leur configuration nécessitent une expertise approfondie, ce qui complexifie leur automatisation complète.
- Protection incomplète : Un IDS seul ne garantit pas une couverture exhaustive contre toutes les cybermenaces et doit être couplé à d'autres mécanismes de sécurité (pare-feu, antivirus, SIEM, etc.).
- Limites face aux attaques avancées : Certains types de menaces sophistiquées, notamment les attaques polymorphes et zero-day, peuvent échapper aux IDS traditionnels.
- Taux d'alerte imparfait : Le rapport entre fausses alertes et attaques non détectées reste un défi majeur, nécessitant une optimisation continue des règles de détection.

8. Présentation des IDS les Plus Répandus

Plusieurs solutions IDS ont été développées, chacune se distinguant par son approche de détection, sa méthode d'implémentation, et son mode de distribution (open-source, propriétaire, commercial, etc.).

8.1. Snort

Développé en 1998 par Martin Roesch [23], Snort est le NIDS open-source le plus utilisé au niveau mondial. Il repose sur une approche par signature et permet aux utilisateurs d'intégrer leurs propres règles de détection. Snort peut fonctionner sous trois modes distincts [24, 25] :

- Mode Sniffer : Capture et affiche en temps réel les paquets réseau transitant sur l'interface surveillée.
- Mode Logger : Enregistre le trafic observé pour une analyse ultérieure.
- Mode IPS : Analyse le trafic en temps réel et applique des mesures correctives en cas de détection d'une menace.

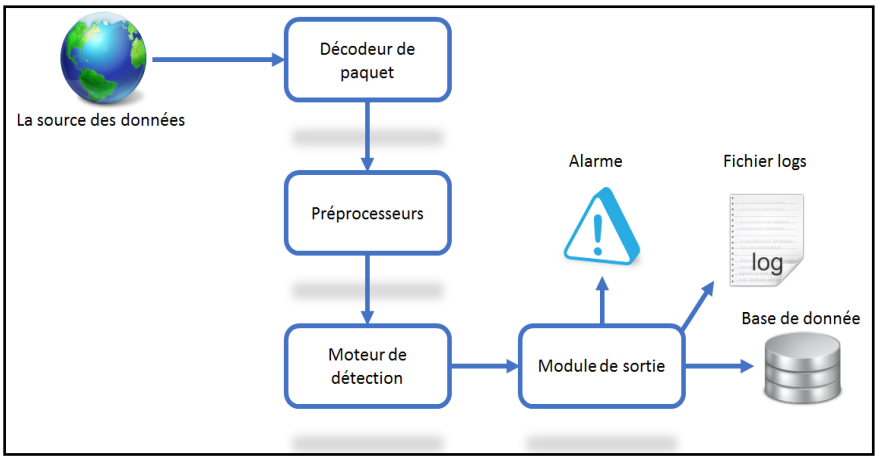

Figure 3 : Architecture de Snort.

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

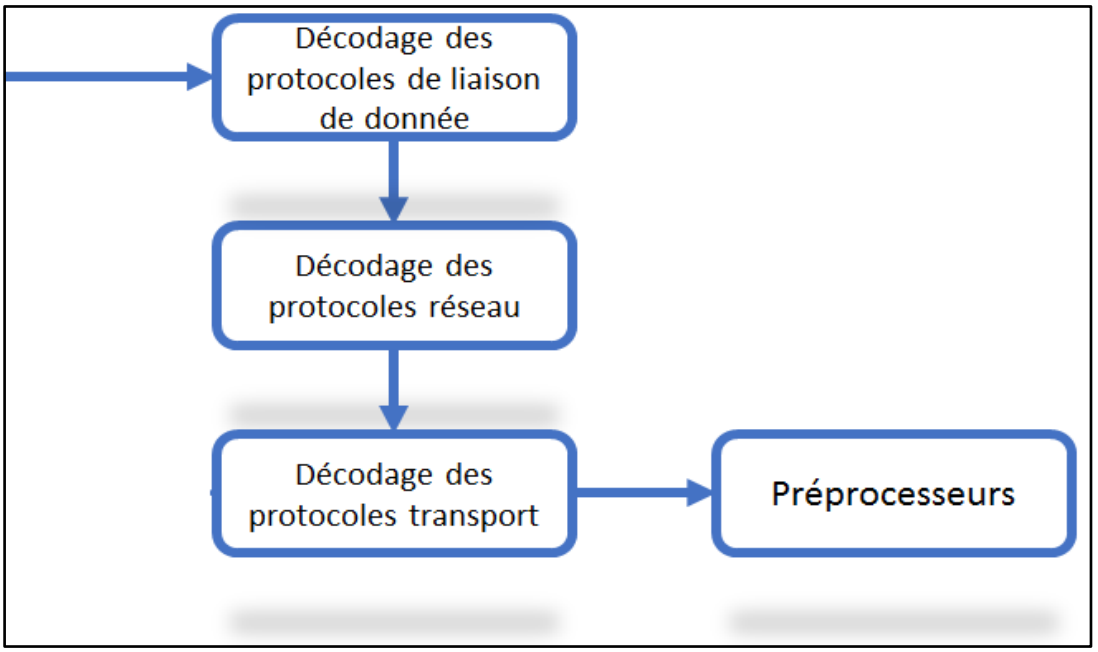

Figure 4 : Le décodeur de paquets

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

L'architecture de Snort est constituée de quatre composants clés :

- Le décodeur de paquets : Identifie et normalise les paquets en fonction des protocoles réseau utilisés.
- Les préprocesseurs : Analyset les paquets pour détecter **des** techniques d'évasion utilisées par les attaquants.

- Le moteur de détection : Compare le trafic observé avec une base de signatures d'attaques connues et déclenche des alertes en cas de correspondance.
- Le système d'alertes et de logs **:** Notifie l'administrateur en cas d'incident et stocke les événements suspects dans des journaux analytiques.

8.2. Prelude

Créé en 1998 par Yoann Vandoorselaere**,** Prelude est un IDS hybride open-source intégrant des fonctionnalités avancées de corrélation des événements. Il combine plusieurs techniques de surveillance [24] :

- Prelude-NIDS : Analyse du trafic réseau en appliquant des règles de détection basées sur des signatures.
- Prelude-LML : Surveillance des fichiers de logs et détection d'anomalies dans les événements système.
- Prelude-Manager : Centralisation des alertes dans une base de données sécurisée, généralement MySQL.

Grâce à son interopérabilité, Prelude est souvent utilisé comme une solution de gestion centralisée pour agréger des alertes issues de plusieurs sources de détection.

8.3. IDES (Intrusion Detection Expert System)

Développé entre 1984 et 1986 par Denning et Neuman**,** IDES est l'un des premiers IDS combinant approche comportementale et approche par signature. Il repose sur deux technologies clés :

- Analyse statistique des comportements : Identification des écarts significatifs par rapport à un profil normal d'utilisation.
- Systèmes experts : Modélisation des menaces connues sous forme de règles de corrélation.

Évolution vers NIDES (Next-Generation IDES, 1993)NIDES a amélioré IDES en introduisant une architecture décentralisée, où les analyses sont effectuées sur une machine distincte du système surveillé, réduisant ainsi l'impact sur les performances de ce dernier [26].

9. Conclusion

Dans ce chapitre, nous avons exploré de manière approfondie les principes fondamentaux de la sécurité informatique, en mettant en évidence les différentes catégories de cyberattaques, leurs classifications ainsi que les stratégies de détection mises en œuvre à travers les systèmes de détection d'intrusions (IDS). Nous avons également étudié les méthodes de fonctionnement et d'implémentation de ces systèmes, en distinguant les approches réactives et proactives, ainsi que les techniques fondées sur la détection par signature et celles basées sur l'analyse comportementale.

Les systèmes de détection d'intrusions sont aujourd'hui une composante essentielle de toute architecture de cybersécurité. Ils permettent d'identifier et de signaler des activités suspectes qui pourraient compromettre l'intégrité des infrastructures informatiques. Toutefois, malgré leur importance indéniable, ces dispositifs présentent certaines limitations qui entravent leur efficacité et leur capacité à offrir une protection complète contre les menaces émergentes. Parmi ces contraintes, nous pouvons citer l'incapacité à prévenir directement les attaques, leur rôle étant essentiellement limité à la détection et à la notification. De plus, la gestion des faux positifs et faux négatifs reste un défi majeur, car un taux élevé de fausses alertes peut submerger les administrateurs de sécurité, tandis que certaines attaques sophistiquées peuvent ne pas être détectées à temps.

Un autre défi réside dans la nécessité d'une maintenance continue et d'une mise à jour fréquente des signatures d'attaques, rendant ces systèmes dépendants d'une surveillance humaine constante. Cette contrainte est d'autant plus problématique face aux attaques de type zero-day et aux menaces polymorphes qui évoluent rapidement et peuvent contourner les mécanismes de détection traditionnels. Par ailleurs, l'intégration de multiples IDS dans une infrastructure réseau complexe peut poser des problèmes d'interopérabilité et d'adaptabilité, rendant difficile la collaboration efficace entre différents systèmes de détection.

Pour pallier ces insuffisances, plusieurs approches ont été envisagées afin de renforcer la coopération entre les IDS et d'améliorer leur capacité d'adaptation aux environnements modernes et distribués. Parmi ces solutions, l'architecture orientée services représente une approche prometteuse en raison de sa flexibilité et de sa capacité à intégrer des systèmes hétérogènes. Elle permet d'unifier la communication entre plusieurs IDS, d'optimiser la corrélation des événements et de rendre le processus de détection plus dynamique et réactif face aux menaces en constante évolution.

Dans le chapitre suivant, nous approfondirons l'architecture orientée services et son application aux systèmes de détection d'intrusions. Nous examinerons les concepts fondamentaux de cette approche, les technologies sous-jacentes qui la soutiennent ainsi que les mécanismes qui permettent d'améliorer l'interopérabilité et la flexibilité des IDS. analyserons également les avantages et les limites de cette architecture dans un contexte decybersécurité, tout en mettant en lumière les défis liés à sa mise en œuvre dans des infrastructures complexes.

Chapitre -2-

L'Architecture Orientée Services (SOA) et les Services Web

1. Introduction

Avec l'essor des systèmes d'information distribués et la nécessité croissante d'interopérabilité dans des environnements de plus en plus complexes et hétérogènes, une nouvelle approche architecturale a émergé : l'architecture orientée services (SOA). Cette dernière repose sur la décomposition des applications en services autonomes, faiblement couplés et accessibles via des interfaces standardisées, ce qui permet d'assurer une flexibilité et une évolutivité accrues.

L'architecture orientée services vise à répondre aux besoins des entreprises en matière d'intégration d'applications et de mutualisation des ressources informatiques. Elle repose sur la mise en œuvre de principes fondamentaux permettant une plus grande modularité des applications et une meilleure interopérabilité entre les systèmes hétérogènes. Grâce à cette architecture, les entreprises peuvent bâtir des infrastructures évolutives, capables de s'adapter aux exigences du marché et aux innovations technologiques.

L'implémentation la plus répandue de SOA repose sur les services web, qui jouent un rôle fondamental dans l'intégration des applications en réseau. Grâce à des protocoles ouverts et à une interopérabilité garantie par des standards comme SOAP, REST, WSDL et UDDI, les services web permettent aux entreprises d'échanger des données et de coordonner des processus métier de manière efficace et évolutive. Ces technologies permettent d'intégrer des applications existantes et d'assurer une communication fluide entre différentes entités logicielles indépendamment de leur localisation, de leur langage de développement ou de leur système d'exploitation.

Ce chapitre s'attache à présenter en détail l'architecture SOA, ses principes fondamentaux, son modèle organisationnel et ses bénéfices. Ensuite, une analyse approfondie des services web, de leurs caractéristiques, de leurs standards et de leur architecture sera fournie. Enfin, nous examinerons les techniques de composition des services, en mettant particulièrement l'accent sur le formalisme ECA, qui confère une flexibilité essentielle aux processus dynamiques et à l'adaptabilité des systèmes d'information face aux changements des exigences métier.

2. L'Architecture Orientée Services (SOA)

L'architecture orientée services est une approche qui privilégie la conception d'applications sous forme de services modulaires et interopérables, facilitant ainsi leur

intégration, leur réutilisation et leur évolutivité [27]. Contrairement aux architectures monolithiques, SOA favorise un faible couplage entre composants, permettant une adaptation rapide aux évolutions technologiques et aux nouveaux besoins métier. Chaque service est un module autonome qui offre des fonctionnalités d'entreprise standard tout en étant indépendant de l'état ou le contexte des autres services, il assure un ensemble défini de fonctionnalités [28]. Les services peuvent être développés dans différents langages de programmation et hébergés sur diverses plates-formes.

2.1. Définition et Principes Fondamentaux

Un service est une entité logicielle encapsulée, offrant des fonctionnalités bien définies accessibles via un contrat d'interface standardisé. Il est caractérisé par plusieurs propriétés fondamentales :

- Autonomie : Chaque service est indépendant et peut être consommé sans nécessiter une connaissance de son implémentation sous-jacente.
- Interopérabilité : Les services utilisent des standards ouverts et peuvent communiquer via divers protocoles, tels que HTTP, SOAP ou REST.
- Faible couplage : Une interaction minimale entre services est privilégiée afin de faciliter leur maintenance et leur évolutivité.
- Découverte dynamique : Les services sont publiés dans des annuaires de services, permettant aux consommateurs de les découvrir et de les invoquer dynamiquement.

Contrairement aux architectures traditionnelles fortement couplées, SOA adopte une approche orientée processus métier, un processus métier est « un ensemble d'une ou plusieurs procédures ou activités liées entre elles pour réaliser collectivement un objectif ou une politique métier en définissant les rôles et les interactions fonctionnelles au sein d'une structure organisationnelle. » [31]. Dans SOA chaque service représente une brique fonctionnelle spécialisée, pouvant être orchestrée et combinée pour former des applications plus complexes.

2.2. Modèle Organisationnel de SOA

L'architecture SOA repose sur trois acteurs principaux :

1. Le fournisseur de services : Acteur responsable du développement, du déploiement, de l'exécution et de la maintenance des services exposés.

2. Le consommateur de services : Acteur qui interagit avec les services en fonction de ses besoins.

Les fournisseurs et les consommateurs sont initialement indépendants, c'est-à-dire que le fournisseur, lors de l'implémentation de son service, n'a pas de connaissances préalables sur ses futurs consommateurs ni sur la manière dont ils réutiliseront son service.

3. L'annuaire de services : Acteur associé à un registre de services.Il joue un rôle central en permettant aux fournisseurs la publication de leurs services et aux consommateurs la recherche et la découverte des services disponibles.

L'interaction entre ces entités repose sur un modèle contractuel défini par des spécifications fonctionnelles et non fonctionnelles, garantissant la qualité et la conformité des services fournis.

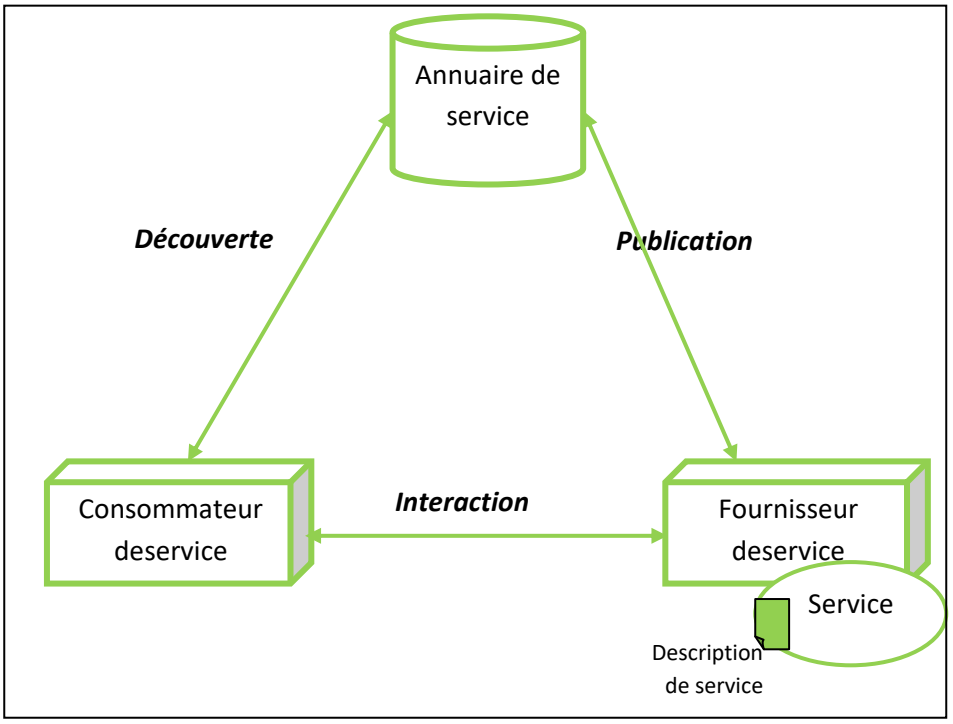

Figure 5 : Organisation d'une architecture orientée services

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies.

2.3. Avantages et Défis de SOA

L'architecture SOA apporte plusieurs bénéfices stratégiques, notamment :

- Réduction des coûts grâce à la réutilisation des services existants et à la consolidation des applications.

- Flexibilité accrue réduisant la complexité des systèmesetfacilitant l'intégration des services et l'adaptation aux évolutions technologiques et métiers.
- Automatisation des processus métier, améliorant la productivité et l'efficacité opérationnelle.
- Interopérabilité optimisée, favorisant l'intégration des applications hétérogènes.

Toutefois, la mise en œuvre de SOA implique des défis majeurs :

- Complexité de gouvernance, nécessitant une gestion rigoureuse des contrats de services et des versions.
- Performance et scalabilité, notamment face à un nombre élevé d'appels inter-services.
- Sécurité des services, avec la nécessité de protéger les communications et d'assurer la confidentialité des données échangées.

3. Services Web : Concepts et Standards

Les services web sont une technologie clé pour l'implémentation de SOA, permettant la communication asynchrone entre applications distribuées via des protocoles standardisés.

3.1. Définition et Caractéristiques des Services Web

À l'origine, la technologie des services web a été initiée par IBM et Microsoft, puis partiellement normalisée sous l'égide du W3C(World Wide Web Consortium), l'organisme chargé de standardiser les évolutions du web. Aujourd'hui, cette technologie est largement adoptée par l'ensemble des acteurs de l'industrie informatique, faisant des services web une technologie révolutionnaire [29].

Un service web est une application logicielle exposée sur un réseau, permettant à d'autres applications de consommer ses fonctionnalités via une interface normalisée.

Les services web présentent les caractéristiques suivantes :

- Standardisation : Ils utilisent des protocoles universels tels que SOAP, REST et WSDL.
- Interopérabilité multi-plateforme : Compatibilité avec divers environnements technologiques.

- Évolutivité : Possibilité d'ajouter de nouveaux services sans affecter les systèmes existants.

3.2. Architecture de services Web

L'architecture des services Web est une instance de l'Architecture Orientée Service (SOA) mentionnée précédemment. Elle définie les éléments globaux qui garantissent l'interopérabilité des services web, permettant à des systèmes hétérogènes de communiquer et de collaborer efficacement.

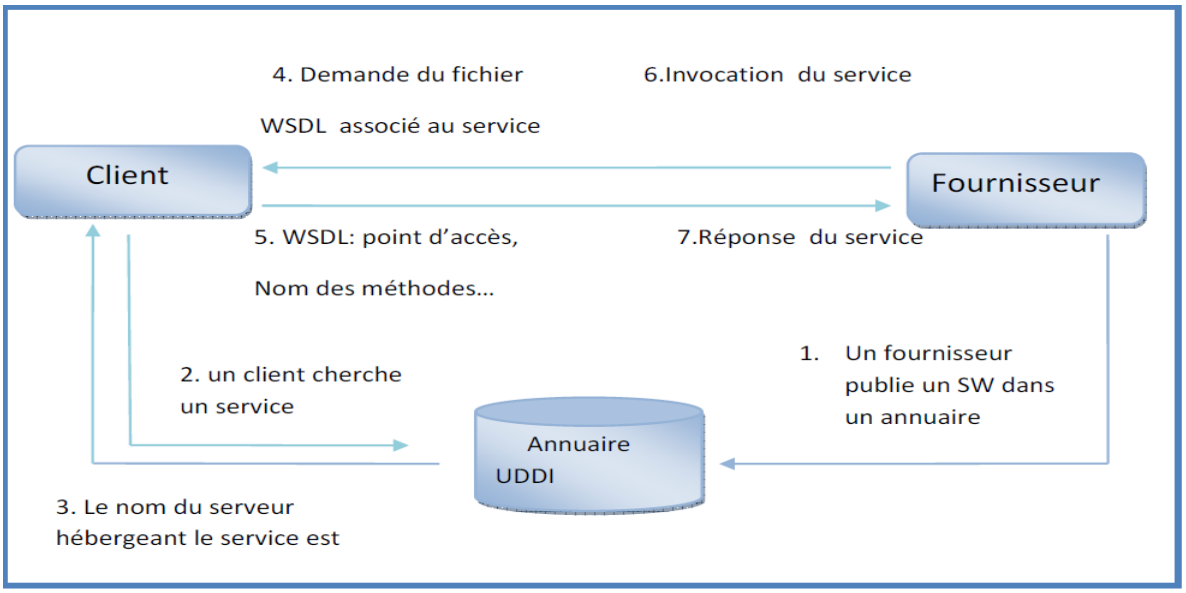

Figure 6 : Architecture de référence des services web

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

Le cycle de vie d'un service Web se déroule en plusieurs étapes :

- **Publication** : Une fois créé, le service web est déployé dans un registre comme l'annuaire **UDDI** (Universal Description, Discovery, and Integration) sur un réseau (local ou Internet).
- **Recherche** : Un utilisateur ayant des besoins spécifiques recherche un service correspondant à ses besoins en interrogeant l'annuaire **UDDI**
- **Lien** : Une fois le service trouvé, l'utilisateur l'invoque et établit une communication avec ce dernier ,les standards utilisé sont le protocole **SOAP**(Simple Object Access Protocol)pour l'échange des messages et le langage **WSDL**(Web Services Description Language)pour la description des services web.

Cette architecture a été étendue en intégrant de couches supplémentaires, telles que celle dédiées à la sécurité et à la composition des services web, afin de répondre à des besoins plus complexes.

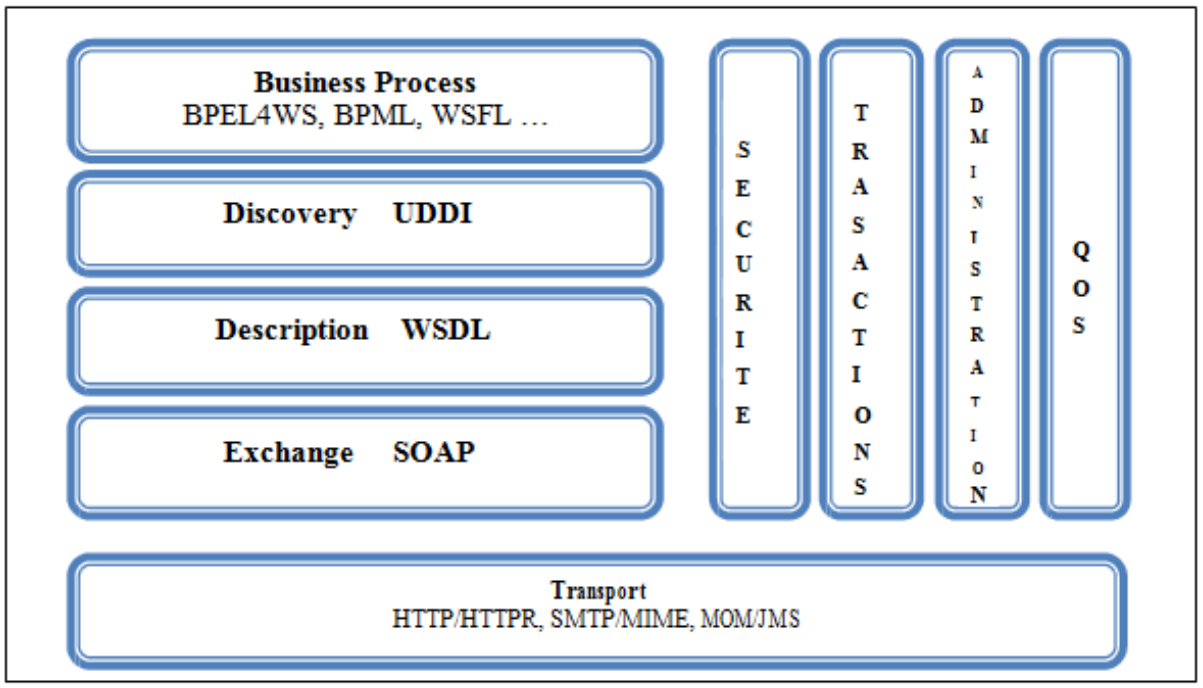

Figure 7 : Architecture étendue des services web.

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

Nous pouvons distinguer dans l'architecture étendue trois types de couches:

- **L'infrastructure de base**: Elle est constituée de trois couches reposant sur les standards émergents SOAP, WSDL et UDDI. Cette infrastructure définit le fondement technique de l'architecture de référence.
- **La couche Processus Métier** : Elle permet l'intégration et l'utilisation efficace des web services dans le domaine du e-business, en alignant les services sur les besoins métier.
- **Lescouches transversales**: Ces couches facilitent l'utilisation opérationnelle des services web dans un contexte industriel, en intégrant des aspects tels que la sécurité, la gestion des transactions et la qualité de service (QoS).

3.3. Standards et Protocoles Associés

Plusieurs standards [30] ont été proposés pour assurer l'interaction entre les participants au sein de l'architecture des services web.

1. SOAP (Simple Object Access Protocol) : Un protocole basé sur XML permettant l'échange de messages structurés entre applications distribuées. SOAP n'est pas lié à un protocole de transport spécifique (bien que HTTP soit couramment utilisé) et il est également indépendant des systèmes d'exploitation et des langages de programmation. Ainsi, en théorie, les clients et serveurs peuvent fonctionner sur n'importe quelle plateforme et être écrits dans n'importe quel langage du moment qu'ils puissent formuler et comprendre des messages SOAP

2. WSDL (Web Services Description Language) : Un langage permettant de décrire l'interface et les opérations des services web de manière standardisée. Son objectif est de fournir une description en XML indépendante de la plateforme et du langage, sous une forme interprétable par des humains ou des programmes. Cette description inclut le point d'accès du service, le type de liaison accepté, les fonctionnalités (ou méthodes) offertes par le service, les types de données utilisés dans les messages.

3. UDDI (Universal Description, Discovery, and Integration) : Un annuaire centralisé facilitant l'enregistrement et la recherche des services web. L'annuaire UDDI est interrogeable selon trois facettes principales:

- ***Pages blanches*** : Elles contiennent des informations sur les noms, coordonnées et descriptions des entreprises, ainsi qu'une liste des identifiants permettant de les repérer.
- **Pages jaunes** : Elles incluent la description au format WSDL des services Web déployés par les entreprises ainsi que les informations permettant de les classifier selon les standards industriels normalisés.
- **Pages vertes** : Elles fournissent des informations techniques détaillées sur les services, y compris les descriptions des services et les informations de liaison.

4. Composition des Services Web

La composition des services web désigne la création d'un nouveau service en réutilisant et en combinant des services existants. La composition définit un processus exécutable, externalisé sous la forme d'un nouveau service, dont les activités constitutives sont elles-mêmes des services web.

L'objectif de la composition de service est de créer de nouvelles fonctionnalités en combinant des fonctionnalités offertes par d'autres services existants, qu'ils soient

déjà composés ou non. Cela permet de réduire le coût et le temps de développement des applications basées sur les services Web et d'apporter une valeur ajoutée à l'utilisateur final.

La composition peut être réalisée de deux manières :

- **Composition ad hoc** : Il s'agit d'un assemblage de plusieurs services web, dont les interactions sont codées manuellement par le développeur. Ce dernier est responsable de l'organisation et du déroulement des processus métiers.
- **Composition via des langages dédiés** : Cette approche utilise des langages spécialisés pour définir et orchestrer les interactions entre les services web, offrant ainsi une méthode plus structurée et automatisée pour la composition.

4.1. Catégories de la composition de services

Selon que la sélection des services et la gestion du flux soient effectuées **a priori** ou non, une approche sera qualifiée de statique ou de dynamique.

▪ Approche statique

Une composition est statique lorsqu'elle intervient lors de l'étape de conception, c'est-à-dire au moment où l'architecture et la conception du système logiciel sont définies. Dans cette approche, les composants (ou services) à utiliser sont sélectionnés et interconnectés a priori, et la gestion du flux est planifiée à l'avance [32].

▪ Approche dynamique

Une composition de services est dite dynamique si les services sont sélectionnés et composés à la volée en fonction des besoins exprimés par l'utilisateur [33].

Une approche dynamique pour la composition de services offre la possibilité de réaliser des applications flexibles et adaptables en sélectionnant et en combinant les services de manière appropriée sur la base de la requête et du contexte de l'utilisateur.

4.2. Techniques de composition des services web

Les travaux du domaine [34,35] identifient deux principales techniques de composition : la chorégraphie et l'orchestration.

4.2.1. Orchestration

Dans une orchestration, le chef d'orchestre est le service dont les interactions sont définies. L'orchestration permet d'enchaîner un service à d'autres services d'une manière

prédéfinie. Le moteur d'orchestration agit comme un contrôleur centralisé qui gère l'exécution des services Web impliqués et coordonne l'exécution des différentes opérations des services Web participant au processus. WS-BPEL(Web Services Business Process Execution Language) [36] et ebXML (Electronic Business using XML**)** [37] sont des exemples de langages d'orchestration.

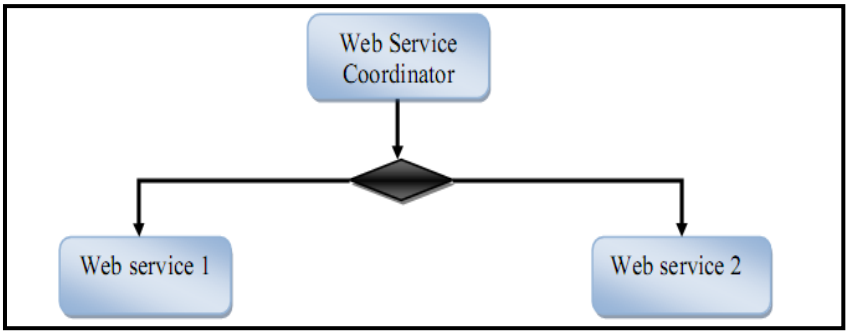

Figure 8 : Orchestration des services web

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

4.2.2. Chorégraphie

Une approche collaborative décentralisée où chaque service connaît son rôle dans l'échange d'informations. Elle permet la modélisation d'un point de vue global afin de prendre en compte des situations de concurrences dans des environnements distribués et ainsi donner une vue plus flexible [38].

Contrairement à l'orchestration, la chorégraphie ne repose pas sur un coordinateur central. Chaque service web impliqué dans la chorégraphie connaît exactement quand ses opérations doivent être exécutées et avec qui l'interaction doit avoir lieu.

Un exemple de langage pour la description de chorégraphies est WS-CDL (Web Services Choreography Description Language) [39].

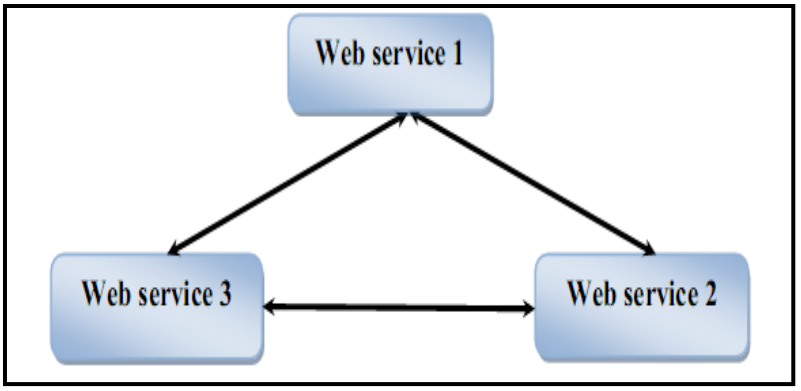

Figure 9 : Chorégraphie de services Web

4.3. Langages de composition

Les langages de composition des services web permettent de spécifier le déroulement des activités d'un processus. Ces langages interprétés par des moteurs d'exécution, utilisent une syntaxe XML pour décrire l'implémentation du processus. Ils peuvent être classés, selon leur origine, en deux familles principales, selon leur origine :

- **Langages de workflow** : Cette famille de langages est destinée à modéliser, au sein d'une entreprise, les flux d'informations échangés entre les différents acteurs et les activités à accomplir par ces acteurs. Parmi les langages les plus connus de cette famille, on trouve XPDL (XML Process Definition Language) [40], ebXML (Electronic Business using XML) [37, 41], BPML (Business Process Modeling Language)[46],

- **Langages d'orchestration** : Cette deuxième famille de langages d'exécution de processus est conçue pour orchestrer les services web impliqués dans un processus métier. Ces processus métier peuvent alors être considérés comme des services web complexes. Les langages d'orchestration exploitent les capacités d'extension du WSDL [42]. Ainsi, leur utilisation ne remplace pas le WSDL, mais vient compléter la description des interfaces exprimées en WSDL, en ajoutant des informations sur les flux de contrôle et les interactions entre les services. . Parmi les langages les plus connus de cette famille, on trouve WSFL (Web Services Flow Language) [43], XLANG [44], et BPEL4WS (Business Process Execution Language for Web Services) [45, 47].

4.4. Les langages de composition basés sur les règles

De nombreuses langages de règles ont été proposés, et par conséquent, plusieures classifications de règles sont apparues [48, 49]. La classification de **Wagner** [50] est la plus largement référencée. Cette classification distingue cinq catégories de règles métier :

- **Les règles d'intégrité :** Il s'agit de contraintes ou d'assertions qui doivent être satisfaites en permanence. Elles garantissent la cohérence et la validité des données.

- **Les règles de dérivation :** Ces règles associent une ou plusieurs conditions à une ou plusieurs conclusions. Elles sont utilisées pour inférer de nouvelles informations à partir de données existantes.

- **Les règles de production** : Ces règles associent une ou plusieurs conditions à une ou plusieurs actions.

- **Les règles de réaction : Basées sur le formaisme ECA,** Ces règles sont déclenchées par des occurrences d'événements et exigent la satisfaction de conditions pour exécuter des actions.
- **Les règles de transformation :** Ces règles contrôlent les changements d'état du système.

4.5. Formalisme ECA (Event-Condition-Action)

Les règles ECA sont des extensions des règles de production (règles de type **Condition-Action**, ou **CA**. Les règles ECA permettent de modéliser des interactions réactives, structurées sous la forme suivante :

ON (Événement) → IF (Condition) → DO (Action)

- **Événement (Event)** : Détermine quand une règle doit être évaluée.
- **Condition (Condition)** : Un prédicat qui détermine si l'action doit être exécutée. Elle peut être vue comme un affinement de l'événement.
- **Action (Action)** : Spécifie le code à exécuter si la condition est satisfaite.

La sémantique attachée à une règle ECA est la suivante : lorsqu'un **événement** se produit, la **condition** est vérifiée. Si la condition est vraie, l'**action** est exécutée en tenant compte des attributs associés à la règle.

Selon plusieurs travaux, [51], [52], [*53*] et [*54*], les règles de réaction (basées sur le formalisme **ECA**) sont les mieux adaptées pour décrire la logique du processus à travers un ensemble de règles. Girrca et al. Dans [*54*], justifient cela par le fait que le formalisme ECA permet de spécifier le flux de contrôle d'un processus d'une manière flexible en utilisant les événements. De plus, ces règles sont faciles à maintenir et permettent d'intégrer tous les types de règles (contraintes, déviations, productions, et transformations) [*53*].L'approche des règles ECA est particulièrement efficace dans les environnements distribués et dynamiques, nécessitant une adaptation rapide aux changements contextuels.

5. Conclusion

Ce chapitre a exploré en profondeur l'architecture SOA, ses principes fondamentaux, ainsi que son implémentation via les services web. Nous avons mis en évidence les standards et protocoles associés, ainsi que les différentes stratégies de composition des services. Enfin,

nous avons étudié le formalisme ECA, qui joue un rôle clé dans la gestion dynamique et adaptative des interactions entre services.

L'adoption de SOA et des services web a profondément transformé la manière dont les systèmes d'information sont conçus et intégrés, offrant une meilleure flexibilité et une interopérabilité accrue. Toutefois, l'évolution constante des besoins métier et des menaces en cybersécurité implique une gestion rigoureuse des services et de leur gouvernance. Dans le chapitre suivant, nous nous concentrerons sur l'application de SOA dans la détection d'intrusions, illustrant comment cette architecture permet d'améliorer la coopération et l'efficacité des systèmes de cybersécurité.

Chapitre -3-

Vers une Détection d'Intrusion Coopérative et Flexible Basée sur une Approche Orientée Services

1. Introduction

Ces dernières années, les systèmes de détection d'intrusions (IDS) se sont imposés comme des mécanismes incontournables dans les stratégies de cybersécurité. Leur déploiement s'est généralisé dans les infrastructures critiques et les entreprises soucieuses de protéger leurs systèmes d'information. Comme évoqué dans le chapitre précédent, chaque type d'IDS présente des avantages et des limites intrinsèques, résultant de la technique de détection employée et du paradigme de conception adopté. Pour pallier ces insuffisances, l'intégration de plusieurs IDS complémentaires s'est avérée une solution efficace, permettant de conjuguer leurs atouts respectifs et de réduire leurs vulnérabilités.

Cependant, cette approche collaborative se heurte à plusieurs obstacles techniques. En effet, les IDS ont été conçus de manière indépendante, sans normalisation des protocoles, des formats de communication ou des modèles d'analyse. Cette hétérogénéité rend complexe leur interopérabilité et entrave une coopération efficace.

Ce chapitre se consacre à l'étude des efforts de standardisation déployés pour favoriser une meilleure interopérabilité entre les IDS. Nous analyserons également les motivations qui sous-tendent l'adoption de l'architecture orientée services (SOA) et des technologies de services web dans le cadre de la détection d'intrusions. Enfin, nous proposerons une synthèse des travaux de recherche ayant contribué à l'application de ces approches au domaine de la cybersécurité.

2. Nécessité d'une détection d'intrusion coopérative flexible

À l'heure actuelle, la majorité des IDS disponibles reposent sur une seule méthode de détection [55]. Or, chacune de ces approches présente des limites intrinsèques.

- Les IDS fondés sur la détection comportementale génèrent un taux élevé de faux positifs, car toute divergence par rapport à un modèle de normalité est interprétée comme une intrusion potentielle. Ces alertes erronées mobilisent des ressources analytiques considérables et peuvent détourner l'attention des véritables menaces.
- Les IDS basés sur la détection par signature ne permettent pas d'identifier des attaques inédites, car leur efficacité repose sur une base de signatures préalablement établie. Cette approche accroît le risque de faux négatifs et nécessite une mise à jour constante de la base de données des signatures, ce qui constitue une contrainte opérationnelle lourde.

Par ailleurs, chaque IDS intègre des mécanismes de détection spécifiques qui influencent son efficacité. Certains outils sont optimisés pour identifier des menaces locales, tandis que d'autres sont conçus pour surveiller des réseaux étendus et détecter des attaques distribuées sophistiquées. Dans ce contexte, le partage d'informations entre IDS devient essentiel, permettant aux systèmes de s'alerter mutuellement sur les menaces émergentes [56, 57].

L'objectif de cette coopération est double :

1. Améliorer la détection des intrusions en exploitant des méthodes complémentaires.
2. Réduire le taux de faux positifs et de faux négatifs grâce à la corrélation des alertes issues de plusieurs sources.

Toutefois, la mise en œuvre d'une telle coopération suppose une capacité d'adaptation en temps réel. Les IDS doivent être en mesure de reconfigurer dynamiquement leurs paramètres en fonction des menaces détectées et des conditions environnementales. Or, les solutions actuelles manquent de flexibilité et d'extensibilité, rendant leur évolution difficile [58].

Face à ces défis, des travaux de standardisation ont été initiés afin de structurer les échanges d'informations entre IDS et d'assurer leur interopérabilité.

3. Efforts de standardisation et interopérabilité des IDS

L'interopérabilité entre les divers systèmes de détection d'intrusion (IDS) constitue un défi majeur dans le domaine de la cybersécurité. La complexité de cette interopérabilité réside principalement dans le fait que la majorité des IDS actuels reposent sur des paradigmes de développement hétérogènes, adoptent des protocoles de communication distincts et implémentent des architectures variées. En conséquence, l'échange d'informations et la coopération entre ces systèmes demeurent limités et non standardisés, réduisant ainsi l'efficacité des stratégies de défense contre les cyberattaques distribuées et multivectorielles. Face à cette problématique, divers projets de recherche ont été créés le développement des formats standards d'échange d'information entre plusieurs systèmes de détection d'intrusions. Les travaux les plus aboutis sont ceux du groupe CIDF et IDWG de l'IETF.

- **CIDF (Common Intrusion Detection Framework)**

En 1997, DARPA(Defense Advanced ResearchProjects Agency, USA)a initié le projet de recherche CIDF (Common Intrusion Detection Framework) dans le but de coordonner les différents projets financés par DARPA et assurer l'interopérabilité entre les outils qui en résultent [59]. Les développeurs de ce projet ont mis en place un modèle permettant l'interopérabilité entre les différents composants d'un système de détection d'intrusions. Cet effort a été complété par le langage CISL (Common Intrusion SpecificationLanguage) qui assure la représentation et la communication des données entre ces composants. Ce langage n'a pas été adopté par la plupart des industriels, ce qui a mis fin au projet en 1999.

- **IDWG**

L'IETF (*Internet Engineering Task Force*) est l'organisation qui s'occupe du développement de nouveaux standards Internet et suite à L'échec du CIDF, Elle a crée un groupe de travail, nommé IDWG (*Intrusion Detection Working Group*) chargé de définir les formats des données et les procédures d'échange destinées à permettre aux IDS de partager des informations, et éventuellement, d'interagir avec des systèmes de gestion au besoin[60]. Ce groupe de travail a proposé un format pour les messages échangés nommé IDMEF (Intrusion Detection Message Exchange Format), qui est une spécification XML pour le format des alertes, ainsi qu'un protocole de communication : IDXP (Intrusion Detection eXchange Protocol)[61]. Le groupe IDWG a aussi proposé une architecture typique d'un système de détection d'intrusions.

4. Application de l'orienté service dans la détection d'intrusion coopérative

La recherche a progressivement orienté ses efforts vers des approches basées sur l'architecture orientée services (SOA) afin de proposer un cadre modulaire, flexible et interopérable pour la détection d'intrusions coopérative. Plusieurs études ont démontré la pertinence de l'intégrationdes services web dans la gestion des IDS, permettant ainsi d'uniformiser les communications et d'améliorer la capacité d'adaptation des systèmes [55, 58, 62, 63, 64].

Dans cette section, nous analysons les motivations principales qui justifient l'adoption de SOA dans le domaine de la cybersécurité, puis nous présentons un état de l'art des travaux ayant appliqué cette approche à la coopération entre IDS.

4.1. Motivation de l'application de SOA dans la détection d'intrusions

L'architecture orientée services (SOA) repose sur un modèle où chaque fonctionnalité logicielle est encapsulée sous forme de services autonomes, accessibles via des interfaces bien définies. Ce paradigme permet une communication fluide et flexible entre des entités logicielles hétérogènes, facilitant ainsi l'interopérabilité entre plusieurs IDS, indépendamment de leurs différences architecturales.

Les avantages majeurs de l'adoption de SOA pour la détection d'intrusion sont les suivants :

- Interopérabilité multi-plateforme : Les services web assurent une communicationnormalisée et indépendante des technologies sous-jacentes, permettant l'interconnexion fluide de divers IDS fonctionnant sur des infrastructures variées.
- Modularité et réutilisation : L'encapsulation des fonctionnalités IDS sous forme de services indépendants favorise la réutilisation des composants, réduisant ainsi les redondances et améliorant la maintenance des systèmes de sécurité.
- Orchestration et chorégraphie avancées : L'organisation des services IDS peut être optimisée à travers des mécanismes de composition dynamique, rendant possible une adaptation en temps réel des stratégies de détection face à l'évolution des menaces.
- Adaptabilité et extensibilité : Grâce à une architecture faiblement couplée, les IDS peuvent être mis à jour ou remplacés sans perturber l'ensemble du système de détection, garantissant ainsi une évolutivité constante.
- Amélioration de la gestion des alertes : L'utilisation de services web facilite la corrélation des événements de sécurité et permet d'établir des mécanismes de réponse coordonnés entre plusieurs IDS.
- Facilitation de l'intégration avec d'autres mécanismes de sécurité : SOA permet de connecter les IDS à d'autres dispositifs de cybersécurité (pare-feu, systèmes de prévention des intrusions, gestionnaires d'événements de sécurité, etc.), créant ainsi une infrastructure de défense unifiée et automatisée.

- Évolutivité face aux attaques complexes : Dans un contexte où les attaques sont de plus en plus distribuées et sophistiquées, l'utilisation de services interconnectés permet de mieux détecter les comportements anormaux et d'anticiper les nouvelles menaces.

Ainsi, SOA s'impose comme une alternative stratégique aux architectures traditionnelles, en garantissant une meilleure flexibilité, une interopérabilité accrue et une capacité d'adaptation optimisée pour la coopération entre IDS.

4.2. Travaux de recherche appliquant SOA à la détection d'intrusions coopérative

Plusieurs travaux de recherche ont exploré l'application des architectures orientées services dans le domaine de la détection d'intrusions coopérative. Ces travaux s'articulent principalement autour de trois axes :

1. La conception d'infrastructures distribuées basées sur SOA pour la coordination des IDS.
2. L'intégration de services web pour la standardisation et l'amélioration de l'efficacité des IDS.
3. L'optimisation des performances des IDS grâce à des approches dynamiques et adaptatives.

[62, 63] : Une approche novatrice pour la composition d'IDS distribués

Dans cette étude, les auteurs proposent une infrastructure de détection d'intrusion distribuée et coopérative permettant d'intégrer divers IDS dans un environnement hétérogène et à grande échelle. L'objectif principal de cette approche est de :

- Combiner des éléments IDS fonctionnant sur différents réseaux pour mutualiser les ressources de détection.
- Assurer un partage efficace des alertes de sécurité en utilisant des services web interopérables.
- Faciliter la coopération entre IDS en adoptant le modèle IDWG comme cadre de référence.

L'infrastructure proposée est basée sur SOA et permet une composition dynamique des IDS, garantissant ainsi une flexibilité accrue pour l'administration de la sécurité. Les services web sont utilisés comme moyens d'interaction standardisés, permettant aux IDS de communiquer efficacement et d'échanger des informations sur les cyberattaques en cours.

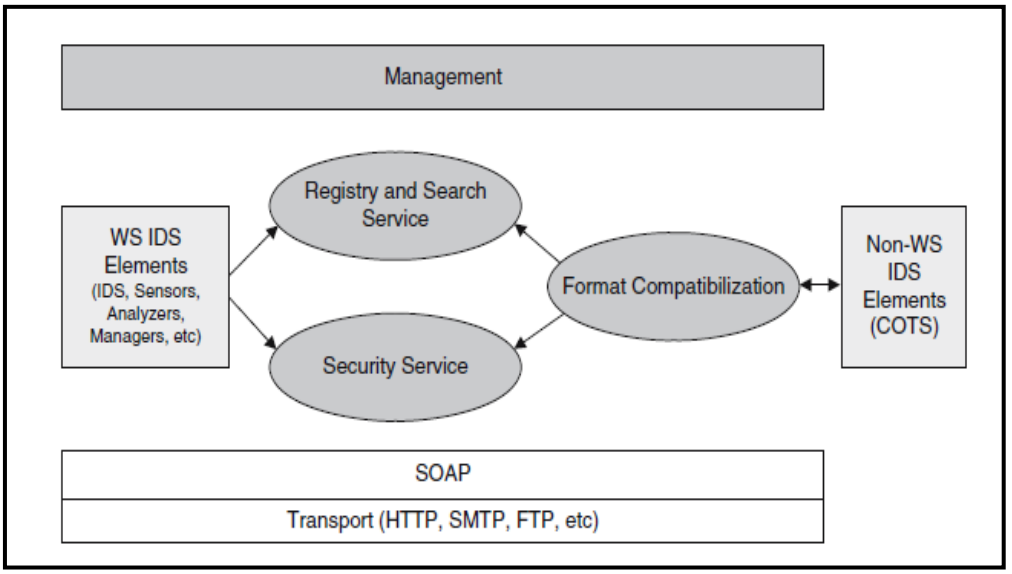

Figure 10 : L'infrastructure des compositions des IDS proposée par [62]

L'analyse des travaux existants met en évidence les bénéfices de l'approche SOA dans la gestion des IDS, notamment en ce qui concerne l'interopérabilité, la flexibilité et l'adaptabilité. Toutefois, plusieurs limitations subsistent :

- La surcharge induite par l'orchestration des services web peut entraîner des latences dans la détection d'intrusions en temps réel.
- Les questions de sécurité et de confidentialité doivent être prises en compte pour éviter que l'échange d'informations entre IDS ne devienne une vulnérabilité exploitable par des attaquants.
- La gestion des politiques de contrôle d'accès et d'authentification dans un environnement SOA nécessite des mécanismes robustes pour éviter les fuites d'informations sensibles.
- L'intégration de techniques d'intelligence artificielle et d'apprentissage automatique pourrait améliorer la capacité de prise de décision et l'efficacité des IDS basés sur SOA.

Dans le chapitre suivant, nous proposerons une approche avancée de coopération inter-IDS basée sur SOA, en intégrant des mécanismes adaptatifs et en optimisant l'orchestration des services de sécurité pour une détection plus efficace des cyberattaques.L'infrastructure proposée dans cette étude vise à intégrer des mécanismes dynamiques pour la composition et la gestion des IDS, en mettant l'accent sur la flexibilité et l'adaptabilité des systèmes de détection d'intrusion. L'objectif est de permettre aux administrateurs de sécurité d'avoir une

supervision plus fine et une réactivité accrue face aux menaces émergentes. Afin d'atteindre cet objectif, les auteurs ont mis en œuvre une orchestration de services basée sur SOA, où la composition des services est réalisée à l'aide de BPEL (Business Process Execution Language). Ce choix technologique permet de modéliser et automatiser les interactions entre différents IDS à travers un ensemble de processus métier prédéfinis, assurant ainsi une coopération fluide et une interopérabilité accrue entre des composants hétérogènes.

[58] : Une approche basée sur les services pour une nouvelle génération de systèmes de détection d'intrusion

Dans cette étude, les auteurs soulignent les limites des IDS classiques, qui sont souvent conçus comme des applications autonomes et isolées les unes des autres. Cette architecture monolithique réduit leur efficacité dans des environnements informatiques collaboratifs, tels que les grilles de calcul et les infrastructures distribuées, qui nécessitent des mécanismes adaptatifs et une communication inter-systèmes fluide. Les chercheurs considèrent que les IDS traditionnels sont insuffisants pour garantir une détection efficace des cyberattaques modernes, car ils souffrent de plusieurs lacunes :

- Absence de flexibilité : Les IDS existants ne disposent pas de mécanismes permettant d'adapter dynamiquement leurs configurations en fonction des nouvelles menaces.
- Manque d'interopérabilité : Ces systèmes utilisent des protocoles distincts et des architectures incompatibles, ce qui limite le partage d'informations et empêche la collaboration entre différents IDS.
- Difficulté d'intégration avec d'autres dispositifs de sécurité : Les IDS actuels ne sont pas conçus pour fonctionner en synergie avec d'autres mécanismes de protection, tels que les pare-feu, les systèmes de prévention d'intrusion (IPS) et les solutions de gestion des événements de sécurité (SIEM).

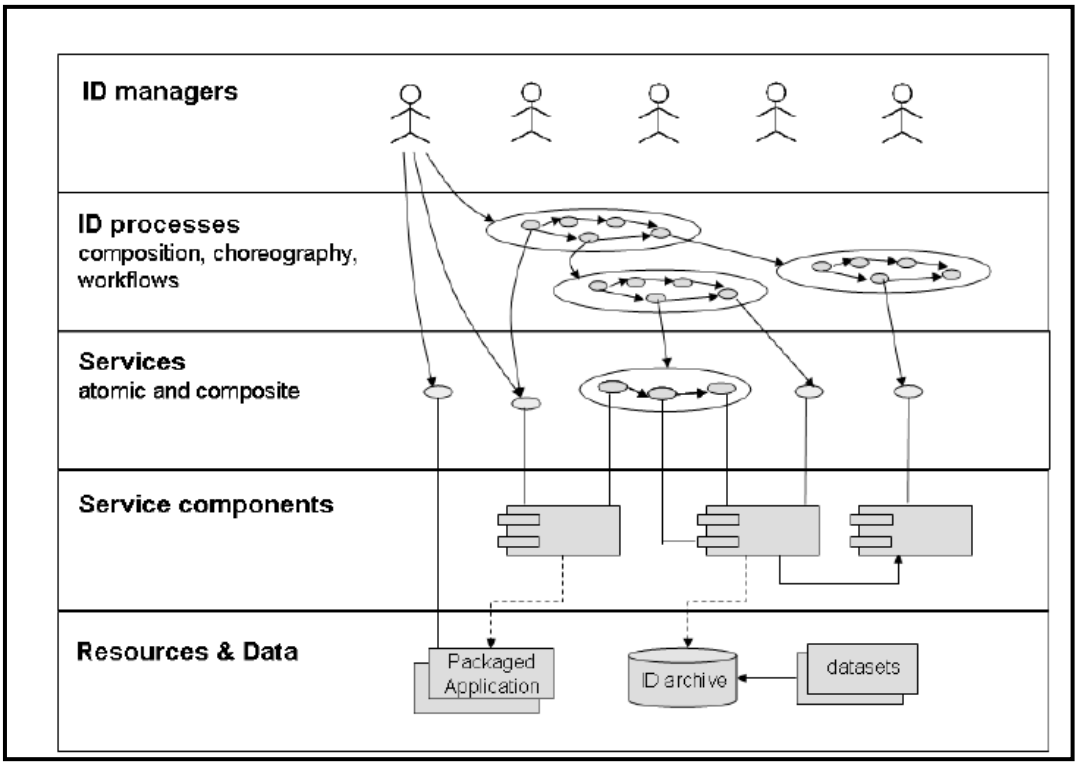

Figure 11 : L'Architecture IDS proposée par [58]

L'infrastructure proposée dans cette étude repose sur l'architecture SOA, permettant une détection d'intrusion distribuée et coopérative. Ce modèle repose sur la composition de plusieurs éléments de détection d'intrusion, qui sont répartis sur différents réseaux et fonctionnent de manière coordonnée.

Cette approche adopte le modèle IDWG, qui définit un cadre structuré pour l'échange de messages et la coopération entre IDS. L'architecture est composée de trois processus fondamentaux, chacun subdivisé en sous-tâches spécialisées dans la détection d'intrusions. Ces processus sont accessibles à travers une infrastructure de grille et s'articulent autour des couches suivantes :

1. La couche des processus d'ID : Cette couche assure une séparation fonctionnelle entre les différentes tâches de détection. Elle permet de modulariser les activités et d'assurer une distribution efficace des charges au sein du réseau surveillé.

2. Le gestionnaire d'ID : Ce composant supervise et orchestre le processus de détection d'intrusion sur un nœud de réseau donné. Il est chargé de collecter, analyser et corréler les informations de sécurité en provenance des capteurs IDS déployés sur l'ensemble de l'infrastructure.

3. Les services de détection distribuée : Chaque service est responsable d'une tâche spécifique, qu'il exécute de manière autonome tout en interagissant avec les autres services pour assurer une analyse approfondie et une détection en temps réel.

Contrairement aux architectures centralisées, où un gestionnaire unique pilote l'ensemble des interactions, cette approche repose sur une composition de services basée sur la chorégraphie.

- Décentralisation des décisions : Chaque service IDS agit indépendamment et partage ses observations avec les autres services de manière asynchrone, favorisant ainsi une réactivité accrue face aux attaques.
- Interopérabilité renforcée : L'usage des services web permet de s'affranchir des différences de plateformes et d'assurer une communication fluide entre IDS hétérogènes.
- Flexibilité et adaptabilité : Grâce à la chorégraphie, il est possible d'ajouter ou de retirer des services IDS dynamiquement, ce qui facilite l'évolution et l'adaptation du système face à des menaces émergentes.

L'intégration de SOA dans les infrastructures IDS présente des avancées majeures en matière d'interopérabilité et d'automatisation, mais elle pose également plusieurs défis techniques qui doivent être pris en compte :

- Optimisation des performances : La mise en place de services distribués peut induire des délais de communication entre les composants IDS, nécessitant ainsi une gestion efficace des flux de données pour éviter toute latence excessive.
- Gestion des politiques de sécurité : L'échange d'informations entre IDS implique un risque potentiel d'exposition aux attaques. Il est donc primordial d'implémenter des mécanismes de contrôle d'accès et de chiffrement robustes.
- Résilience et tolérance aux pannes : L'architecture doit être capable de s'adapter aux pannes de certains services sans compromettre la surveillance globale du réseau.

[65] : Une approche orientée services pour optimiser les performances de Snort

Dans cette étude, les chercheurs proposent une nouvelle architecture orientée services afin d'améliorer considérablement la performance et la flexibilité du système de détection d'intrusions Snort. L'un des défis majeurs des IDS traditionnels réside dans l'efficacité et la précision des algorithmes de détection, qui peuvent varier en fonction des mises à jour et des

caractéristiques des attaques émergentes. Les auteurs mettent en avant la dépendance de Snort à ses algorithmes de détection, en soulignant que certaines versions utilisent des techniques avancées et des heuristiques optimisées, tandis que d'autres reposent sur des algorithmes obsolètes et moins performants. Dans ce contexte, l'objectif de cette approche est de découpler les algorithmes de détection de leur exécution locale en les transformant en services accessibles à distance.L'idée principale de cette architecture repose sur l'externalisation des algorithmes les plus performants vers des serveurs spécialisés, qui les exposent sous forme de services web réutilisables. Cette approche présente plusieurs avantages stratégiques :

- Optimisation des ressources : Plutôt que de surcharger chaque instance de Snort avec des bases de signatures volumineuses, les algorithmes sont hébergés sur des infrastructures centralisées, permettant une exécution plus rapide et efficace.
- Interconnexion mondiale des instances Snort : Chaque système Snort déployé à travers le monde peut invoquer dynamiquement ces services pour bénéficier des algorithmes les plus récents et les plus performants, sans nécessiter une mise à jour manuelle des bases de détection locales.
- Apprentissage adaptatif et enrichissement progressif : Lorsqu'une attaque est détectée via l'un des services distants, les caractéristiques de cette attaque sont stockées localement, permettant ainsi aux instances de Snort d'être plus autonomes sur les menaces déjà identifiées.

Afin de faciliter la coopération entre les différentes instances de Snort et garantir une distribution efficace des algorithmes, les chercheurs ont adopté une approche basée sur des agents mobiles.

- Les agents mobiles sont chargés de collecter les informations relatives aux attaques et de mettre à jour les bases de signatures locales en fonction des algorithmes externes les plus performants.
- L'architecture repose sur une interconnexion dynamique entre les instances Snort et les serveurs d'algorithmes, où chaque requête est acheminée via des protocoles sécurisés garantissant l'intégrité des échanges.
- L'utilisation des services web permet aux instances Snort de s'adapter aux menaces émergentes sans intervention humaine directe, ce qui réduit le temps de réaction face aux nouvelles attaques et améliore la réactivité globale du système.

L'apport principal de cette approche réside dans sa capacité à rendre Snort plus flexible et évolutif, en transformant son moteur de détection en une architecture distribuée et interopérable.

[80] : Vers une interopérabilité accrue des IDS pour la détection d'attaques distribuées en plusieurs étapes

Les auteurs de cette étude soulignent les limites des IDS traditionnels, en particulier leur incapacité à identifier efficacement les attaques coordonnées et planifiées, également appelées attaques multi-étapes. Ces attaques sont souvent orchestrées sur une période prolongée, impliquant plusieurs phases d'intrusion qui, prises isolément, peuvent sembler bénignes ou inoffensives pour un IDS classique. Le principal défi identifié est le manque de communication efficace entre les IDS existants et les autres dispositifs de sécurité, tels que les pare-feu, les systèmes de prévention d'intrusion (IPS) et les solutions SIEM. Cette absence de corrélation des événements rend la détection des attaques complexes extrêmement difficile, car chaque composant du système travaille en silo, sans partage d'informations contextuelles. Pour remédier à ces limites, les auteurs proposent une solution novatrice basée sur les services web. Leur approche repose sur deux contributions majeures :

1. L'introduction d'un langage dédié à la spécification des attaques multi-étapes
2. Le développement d'une architecture orientée services (SECCOMPOSE) capable de détecter et de corréler ces attaques de manière efficace

L'architecture SECCOMPOSE repose sur quatre composants principaux, chacun jouant un rôle spécifique dans la surveillance, la détection et la mitigation des cybermenaces :

- Le module de gestion et de supervision : Cet outil permet aux administrateurs de sécurité de définir des scénarios d'attaques complexes, en spécifiant les différentes étapes d'une intrusion et leurs interactions possibles. Il sert également d'interface d'analyse pour visualiser les alertes générées par les autres composants.
- Le service de notification des événements : Il est responsable de la surveillance continue des activités réseau, en collectant les événements suspects et en les acheminant vers le service de détection. Ce module assure une remontée d'informations en temps réel, ce qui permet une prise de décision rapide.
- Le service de détection avancée : C'est le cœur de l'architecture SECCOMPOSE. Il est chargé d'analyser et de corréler les événements détectés pour identifier les

schémas d'attaques en plusieurs étapes. Grâce à des algorithmes de machine learning et de corrélation d'événements, il permet de mettre en évidence des liens entre différentes actions malveillantes qui, isolément, ne déclencheraient pas d'alerte.

- Le service de contre-mesures et d'atténuation : Une fois qu'une attaque est détectée, ce module déclenche automatiquement des contre-mesures adaptées, telles que le blocage de certaines adresses IP suspectes, la modification des règles de pare-feu ou la mise en quarantaine des machines compromises.

L'approche proposée dans cette étude représente une avancée significative dans la détection des attaques avancées et la gestion des menaces en environnement distribué.

Avantages majeurs de l'architecture SECCOMPOSE

- Détection améliorée des attaques complexes : Grâce à une approche basée sur la corrélation d'événements, cette architecture permet de détecter des attaques sophistiquées qui passeraient inaperçues avec des IDS traditionnels.
- Interopérabilité accrue : En s'appuyant sur les standards des services web, SECCOMPOSE facilite la communication entre les différents IDS et s'intègre facilement avec d'autres solutions de cybersécurité.
- Réactivité et automatisation : L'implémentation d'un module de contre-mesures automatisé permet une réduction significative du temps de réponse en cas d'attaque.

Défis et axes d'amélioration

- Optimisation des performances : La corrélation d'événements en temps réel peut nécessiter une puissance de calcul élevée, ce qui peut affecter la scalabilité du système.
- Fiabilité des détections : La mise en œuvre d'algorithmes d'apprentissage automatique nécessite une base d'apprentissage représentative, afin d'éviter les faux positifs et les faux négatifs.
- Sécurisation des communications : Étant donné que l'architecture repose sur un échange constant d'informations sensibles, il est crucial d'intégrer des mécanismes de chiffrement robustes pour éviter les risques de compromission.

En combinant les capacités analytiques avancées des IDS avec la flexibilité et l'interopérabilité des services web, l'architecture SECCOMPOSE représente une avancée

majeure vers une cybersécurité plus proactive et plus intelligente. Elle ouvre de nouvelles perspectives pour la détection et la prévention des cyberattaques en environnement distribué, tout en offrant une meilleure visibilité et un contrôle accru aux administrateurs de sécurité.

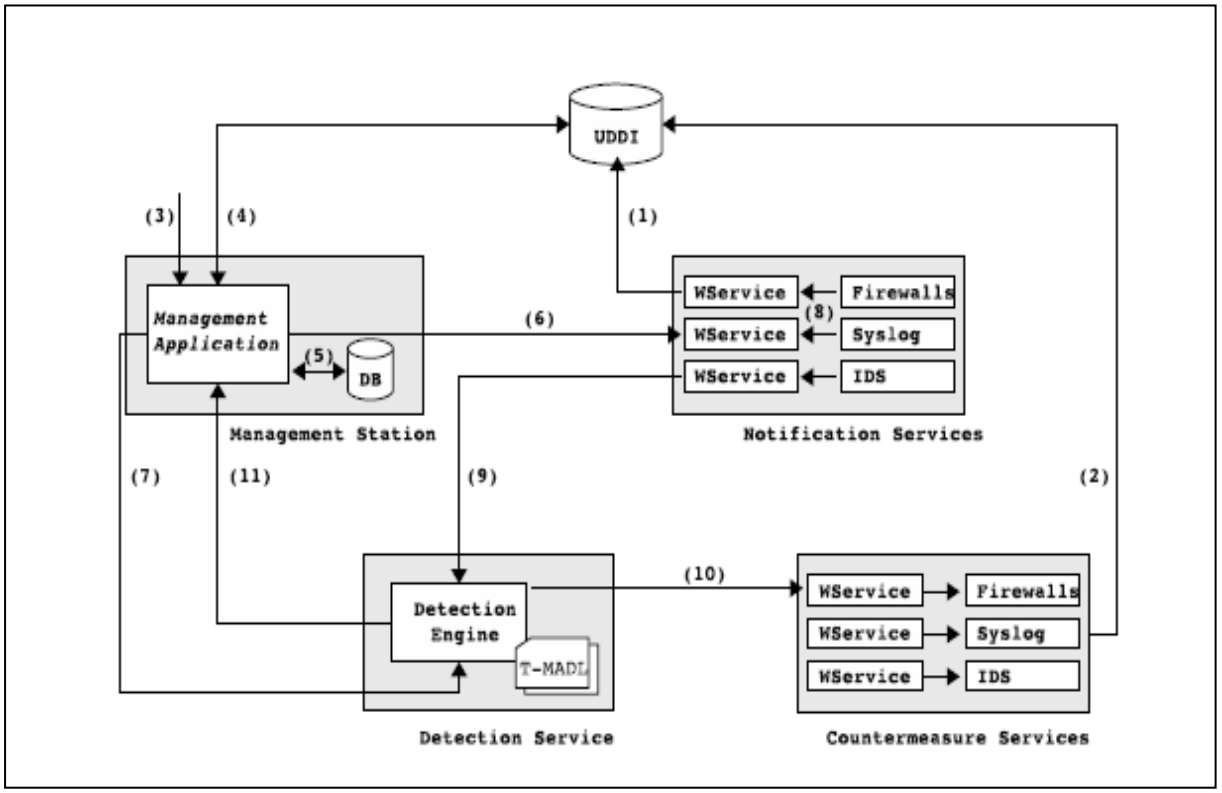

Figure 12 : Les interactions entre les composants de l'architecture proposée par [80]

[55] : Modélisation et analyse orientée services pour la construction des IDS

Dans cette étude, les auteurs mettent en évidence les lacunes des systèmes de détection d'intrusion traditionnels, qui se limitent à une seule approche de détection, ce qui restreint leur capacité à identifier un large éventail de cyberattaques. Pour pallier cette limitation, ils proposent une architecture orientée services, qui vise l'interconnexion et la coopération de plusieurs IDS déployés sur des sites multiples. L'objectif principal de cette approche est de mutualiser les capacités des IDS hétérogènes pour renforcer leur efficacité globale, en facilitant le partage d'informations et la corrélation des alertes dans un environnement distribué.

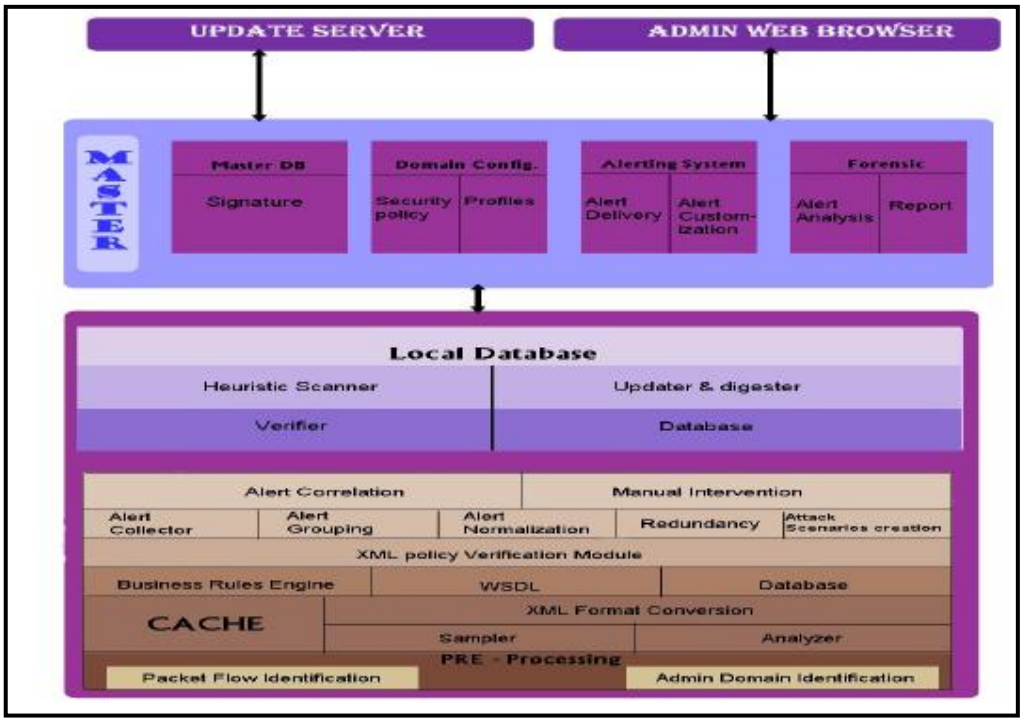

Figure13 : Architecture d'IDS proposée par [55]

L'architecture proposée repose sur plusieurs services spécialisés, chacun assumant un rôle spécifique dans le processus de détection et d'analyse des menaces :

- Pre-Processing : Identifie et segmente les flux de paquets en fonction de leur format (binaire, textuel, etc.), tout en les répartissant selon des domaines administratifs distincts (finance, RH, infrastructure, etc.).
- Sampling&Analyzing : Capture et analyse un sous-ensemble aléatoire de paquets avant de les convertir en format XML, facilitant ainsi leur traitement et leur stockage.
- Détection : Analyse les paquets XML, les compare aux règles métier prédéfinies, et ajuste dynamiquement ces règles en fonction du positionnement de l'IDS dans le réseau.
- Alerting : Gère la génération et la diffusion des alertes en cas d'anomalie détectée, en les adressant aux administrateurs de sécurité concernés.
- Event Recording : Assure la mise à jour dynamique de la base des signatures en s'appuyant sur les données collectées auprès d'autres IDS connectés au réseau.

Cette approche favorise une meilleure coordination entre les IDS multi-sites, en exploitant une architecture modulaire et évolutive, qui permet une réactivité accrue face aux menaces émergentes.

[66] : Système de détection des intrusions réseau basé sur SOA (NIDS-SOA) – Amélioration de l'interopérabilité des IDS

Dans cette étude, les auteurs soulignent l'importance de la qualité de la base des signatures pour l'efficacité des systèmes de détection d'intrusion. Or, les IDS existants sont souvent isolés, ce qui signifie qu'ils ne partagent pas leurs bases de signatures et ne coopèrent pas, rendant ainsi l'apprentissage collectif plus difficile. Pour surmonter cette limitation, ils proposent une approche innovante basée sur l'architecture SOA, baptisée NIDS-SOA, qui vise à assurer une interopérabilité fluide entre plusieurs IDS hétérogènes, en leur permettant d'échanger des informations et des alertes en temps réel.

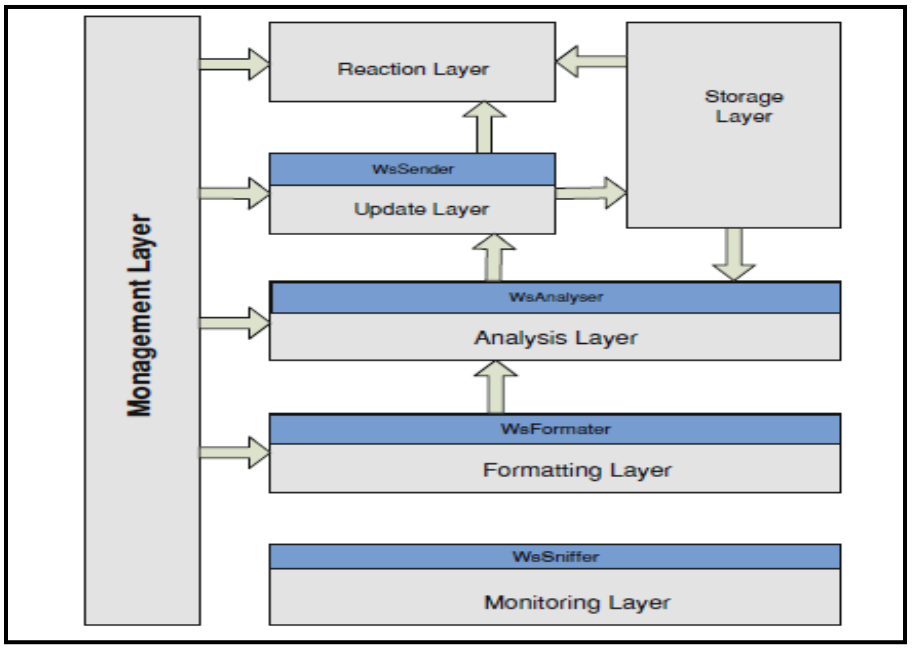

Figure 14 : Architecture de NIDS-SOA [66]

L'architecture proposée repose sur cinq services web interdépendants, chacun assurant une fonction spécifique dans le processus de détection et de réaction aux menaces :

- WsSniffer : Capture les paquets réseau entrants et sortants, en extrayant les informations essentielles pour l'analyse.
- WsFormater : Formate les paquets capturés en fichiers XML, afin de garantir une compatibilité inter-systèmes et faciliter leur traitement par les autres modules.
- WsAnalyser : Analyse les paquets déjà formatés, en détectant les anomalies et les comportements suspects à partir des règles de détection établies.

- WsSender : Met à jour les bases de données des IDS interconnectés, en partageant les nouvelles signatures et alertes détectées.
- WsReaction : Déclenche des actions préventives ou correctives, en fonction des analyses effectuées, telles que le blocage d'une adresse IP suspecte, la mise en quarantaine d'un système compromis ou la réécriture dynamique des règles de pare-feu.

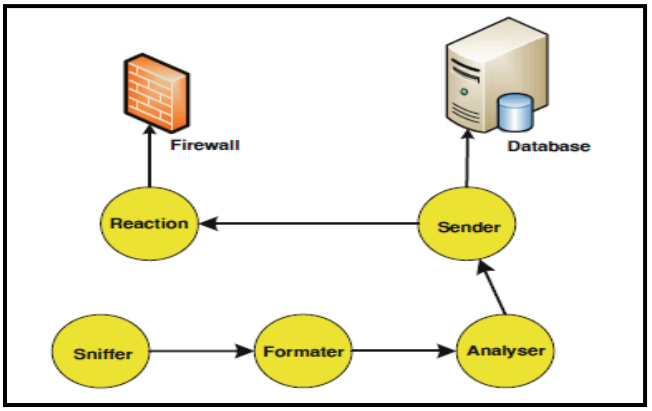

Figure 15 : La composition des services proposée par [66]

L'architecture NIDS-SOA repose sur une stratégie de composition dynamique des services web, qui permet une flexibilité et une évolutivité accrues. Grâce à cette approche, les IDS peuvent coopérer de manière plus efficace, en mutualisant leurs ressources et leurs connaissances, ce qui améliore la rapidité et la précision de la détection des cyberattaques.

[69] : Un service Web RESTful pour les systèmes de détection d'intrusion à grande vitesse

Les chercheurs à l'origine de cette étude s'intéressent aux environnements réseau à haut débit, où les IDS traditionnels peuvent rencontrer des difficultés en raison de la volumétrie des flux de données et de la complexité des attaques modernes. Dans ce contexte, ils proposent une solution basée sur des services web RESTful, qui permet une communication rapide et fluide entre les différents IDS distribués, tout en réduisant les charges de calcul et de stockage.

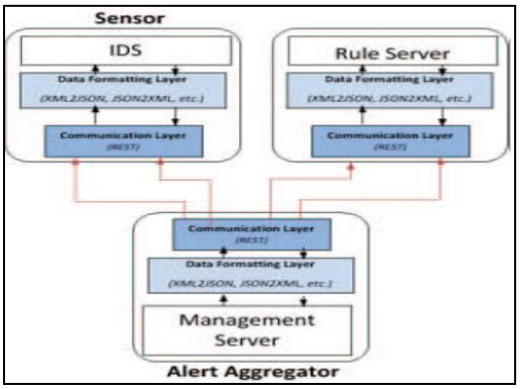

Figure 16 : L'architecture proposée par [69]

L'architecture développée repose sur le modèle IDWG, tout en intégrant une approche RESTful, qui apporte plusieurs bénéfices :

- Légèreté et efficacité : Contrairement aux architectures basées sur SOAP, l'utilisation des API REST permet d'échanger les données de manière optimisée, avec une consommation minimale de ressources.
- Interopérabilité accrue : REST permet aux IDS de s'intégrer facilement avec d'autres dispositifs de sécurité, indépendamment des langages de programmation ou des plateformes sous-jacentes.
- Flexibilité et évolutivité : L'architecture RESTful facilite le déploiement de nouveaux services IDS et leur mise à jour dynamique, ce qui permet d'adapter rapidement le système aux nouvelles menaces.

Dans cette approche, les services web REST sont directement intégrés comme des API dans chaque module du système IDS (capteurs, serveurs de règles, agrégateurs d'alertes). Cette intégration permet une communication en temps réel entre les différents composants, garantissant ainsi une réactivité accrue face aux incidents de sécurité.

Les différentes études analysées montrent l'évolution des IDS vers des architectures plus ouvertes et interopérables, grâce à l'adoption des services web et des modèles SOA. Ces approches permettent d'améliorer considérablement la coopération entre IDS, tout en garantissant une flexibilité accrue et une meilleure évolutivité.

Tableau 02 : Comparaison des approches étudiées

Étude	Approche proposée	Objectif principal	Type d'architecture	Technologie utilisée
[55]	Modélisation orientée services des IDS	Coopération entre IDS multi-sites	SOA	Services web XML
[66]	NIDS-SOA	Améliorer l'interopérabilité entre IDS	SOA	Composition dynamique des services
[69]	RESTful IDS	Détection d'intrusion en réseau à haut débit	IDWG & REST	API REST et services web légers

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

Les résultats obtenus à travers ces travaux montrent que les architectures orientées services constituent une solution prometteuse pour renforcer l'efficacité des systèmes de détection d'intrusions, en particulier dans des environnements hétérogènes et dynamiques.Cependant, certaines limitations persistent, notamment en matière de scalabilité et de gestion en temps réel des événements de sécurité. L'intégration de techniques avancées d'intelligence artificielle et d'apprentissage automatique pourrait permettre d'améliorer encore davantage la précision et la réactivité des IDS.Ainsi, l'avenir des IDS repose sur une hybridation des modèles existants, combinant les avantages des architectures SOA et RESTful avec des capacités avancées de traitement des données en temps réel.

4.3. Synthèse des travaux appliquant SOA à la détection d'intrusions coopérative

L'analyse comparative des approches étudiées dans les travaux précédents met en évidence la diversité des stratégies adoptées pour intégrer SOA dans la détection d'intrusions coopérative. Cette diversité s'exprime à travers plusieurs dimensions clés, notamment :

- L'architecture proposée, qui peut être totalement nouvelle ou basée sur des modèles existants.
- La démarche de modélisation, qui influence la structuration des interactions entre les IDS.

- L'échelle de l'application, permettant d'évaluer si la coopération s'étend à des environnements multi-sites.
- Le type de composition des services pris en charge, allant des solutions statiques aux approches dynamiques basées sur l'orchestration et la chorégraphie.
- L'intégration avec les solutions IDS existantes, un facteur clé pour la compatibilité avec les outils industriels.
- L'existence ou non d'un prototype, permettant de mesurer la maturité et l'applicabilité des solutions proposées.

Tableau 3 : Synthèse des travaux analysés

Travaux	Architecture proposée	Démarche de modélisation	Échelle d'application (multi-sites, inter-entreprises)	Composition des services	Intégration des solutions existantes	Prototype
[55]	Oui	SOMA	Multi	Simple, implicite	Non	Non
[66]	Oui	UML	Multi	Simple, statique	Non	Oui
[80]	Oui	MADL, adaptation de diagramme UML	-	Prédéfinie, statique	Oui	Non
[58]	Oui, basée sur modèle IDWG	-	Grilles	Dynamique	Oui	Non
[65]	Non	-	Multi	-	Non	Oui
[62,63]	Oui, basée sur modèle IDWG	UML	Multi	Dynamique avec BPEL	Oui	Oui
[69]	Oui, basée sur modèle IDWG	-	-	Statique	Non	Non

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

L'analyse des différentes approches met en lumière plusieurs tendances récurrentes et limitations dans l'application de SOA à la détection d'intrusions coopérative :

- De nombreux travaux adoptent une composition statique des services, ce qui limite leur capacité d'adaptation aux menaces émergentes et aux évolutions dynamiques des réseaux.

- L'approche de [62,63] repose sur une composition dynamique utilisant BPEL, ce qui constitue une avancée notable. Toutefois, BPEL étant un langage d'orchestration statique, il ne permet pas une adaptation contextuelle en temps réel, ce qui peut réduire l'efficacité des IDS dans des environnements hautement évolutifs.

- Plusieurs propositions restent théoriques, sans intégration réelle avec les IDS déployés industriellement.

- Seule une minorité des travaux (comme [66]) proposent un prototype opérationnel, ce qui soulève des interrogations sur la faisabilité concrète de certaines approches.

- La majorité des architectures analysées ne prennent pas en charge une reconfiguration dynamique des IDS. Or, les environnements de cybersécurité évoluent rapidement, et un IDS rigide devient obsolète face aux nouvelles menaces.

- Un défi majeur est donc de développer une infrastructure SOA flexible, capable d'ajouter ou de supprimer dynamiquement des services en fonction des besoins de sécurité.

4. Conclusion

Ce chapitre a mis en lumière les défis et les limites des systèmes de détection d'intrusion (IDS) traditionnels, soulignant la nécessité d'une coopération inter-IDS pour améliorer la précision et l'efficacité des mécanismes de détection. Nous avons démontré que, malgré les avancées dans les techniques de détection d'intrusions, les approches existantes restent insuffisantes face aux cyberattaques de plus en plus sophistiquées, notamment celles qui sont distribuées ou évolutives.

L'interopérabilité entre IDS constitue un enjeu central, mais elle est entravée par la diversité des méthodologies de détection, l'hétérogénéité des protocoles de communication et le manque de formats standardisés d'échange d'informations. À cet effet, plusieurs initiatives de standardisation, notamment CIDF et IDWG, ont cherché à établir des protocoles et des formats permettant d'harmoniser la communication entre IDS. Toutefois, ces efforts ont montré leurs limites en raison d'un manque d'adaptabilité et de flexibilité, empêchant ainsi une véritable collaboration dynamique et contextuelle entre IDS.

Dans cette optique, nous avons exploré l'apport des architectures orientées services (SOA) et des services web dans la mise en place d'une coopération inter-IDS flexible et scalable. Contrairement aux approches monolithiques, SOA offre une modularité accrue, permettant aux IDS de collaborer indépendamment de leur technologie sous-jacente. De plus, l'orchestration et la chorégraphie des services facilitent la gestion des processus de détection d'intrusion en automatisant l'échange et l'analyse des alertes.

Les recherches récentes démontrent que l'adoption des services web dans le domaine de la cybersécurité permet d'optimiser l'efficacité des IDS en garantissant une interopérabilitéaccrue, une réduction du nombre de faux positifs et une meilleure capacité d'adaptation aux menaces émergentes. En encapsulant les composants IDS sous forme de services interopérables, il devient possible de concevoir un écosystème collaboratif et évolutif dans lequel les IDS échangent des informations en temps réel et réagissent collectivement aux cyberattaques en cours.

Toutefois, malgré ses nombreux avantages, l'implémentation d'une infrastructure SOA pour les IDS présente encore des défis. Parmi eux, nous pouvons citer :

- Les contraintes de performance, car l'orchestration de services peut introduire une latence supplémentaire dans le traitement des alertes.
- La gestion des politiques de sécurité, notamment en ce qui concerne l'authentification et la confidentialité des échanges entre IDS.
- L'adaptabilité aux environnements à très grande échelle, où le volume des événements de sécurité peut être massif et difficile à corréler efficacement.

Ces limites nécessitent une approche innovante et optimisée pour tirer pleinement parti des bénéfices de SOA tout en assurant une réactivité et une résilience accrues face aux cyberattaques complexes et distribuées.

Dans le chapitre suivant, nous proposerons une nouvelle approche de coopération inter-IDS, en nous appuyant sur les principes de SOA et le formalisme ECA (Event-Condition-Action). Nous détaillerons la conception de notre modèle flexible, son implémentation expérimentale, ainsi que l'évaluation de ses performances dans un scénario de détection d'intrusions en environnement hétérogène et dynamique.

Chapitre- 4-

L'approche Proposée

1. Introduction

L'interconnexion des systèmes de détection d'intrusion (IDS) constitue un levier stratégique pour renforcer la résilience des infrastructures numériques face aux menaces cybernétiques. En effet, la mise en place d'une coopération efficace entre ces dispositifs permet d'optimiser la surveillance, d'accroître la réactivité aux incidents et d'améliorer la capacité d'anticipation des attaques malveillantes. Toutefois, cette synergie demeure insuffisante si elle ne repose pas sur une architecture à la fois adaptable et aisément administrable. L'absence de flexibilité et la complexité de gestion pourraient entraver la prise de décision rapide des responsables de la sécurité et limiter leur aptitude à moduler les réponses défensives en fonction des besoins évolutifs du contexte sécuritaire.

Il apparaît dès lors impératif de concevoir un cadre coopératif intégrant une flexibilité intrinsèque, capable de concilier robustesse des mécanismes de protection et efficacité opérationnelle. Un tel modèle doit non seulement renforcer la résilience face aux menaces, mais également optimiser les délais de détection et de réaction tout en garantissant une interopérabilité fluide avec les exigences des administrateurs de la sécurité informatique.

Dans cette perspective, ce chapitre se propose d'exposer une approche novatrice de coopération inter-IDS, articulée autour d'une architecture orientée services (CIIDS-SOA). Ce cadre conceptuel vise à instaurer un modèle d'échange sécurisé, évolutif et hautement performant entre les différents IDS. Nous détaillerons ainsi les principes fondamentaux de cette architecture et nous illustrerons son implémentation afin d'évaluer son efficacité et sa pertinence dans des environnements à forte criticité sécuritaire.

2. Framework proposé

Le cadre conceptuel que nous proposons pour la coopération entre systèmes de détection d'intrusion (IDS) s'appuie sur les principes de l'architecture orientée services (SOA), l'exploitation des services web et l'implémentation des règles événement-condition-action (ECA). L'adoption de cette approche repose sur la capacité de SOA à pallier les problématiques d'interopérabilité, facilitant ainsi le partage d'informations et la mutualisation des fonctionnalités des IDS au sein d'un environnement dynamique et évolutif.

Dans le domaine de la coopération inter-IDS, plusieurs implémentations de l'architecture SOA ont été explorées. Parmi ces différentes alternatives, notre choix s'est porté sur les services web en raison de leur adoption généralisée tant dans l'industrie que dans

la recherche académique. Leur standardisation, leur compatibilité avec divers protocoles et leur flexibilité en font un vecteur privilégié pour assurer une intégration harmonieuse et évolutive des IDS.

Notre proposition repose également sur une approche innovante de composition dynamique des IDS. Contrairement aux architectures rigides et figées, une stratégie de composition dynamique des services permet d'adapter en temps réel l'infrastructure sécuritaire en fonction des besoins spécifiques de l'utilisateur et du contexte opérationnel. Cette adaptabilité est essentielle pour garantir une gestion efficace et agile de la sécurité. En effet, un administrateur doit pouvoir orchestrer, combiner et réutiliser des services de manière rapide et intuitive afin de répondre aux impératifs métier et aux nouvelles menaces émergentes.

L'un des enjeux majeurs de cette approche est d'accélérer la prise de décision et d'améliorer le temps de réponse du gestionnaire face aux évolutions contextuelles de l'environnement numérique. L'intégration des règles ECA joue ici un rôle déterminant : elles offrent une capacité avancée de modélisation et d'automatisation des scénarios de coopération, assurant ainsi une gestion plus fine et dynamique des interactions entre IDS.

3. Avantages des Règles ECA sur la Composition des Services Web

L'efficacité d'un processus de coopération inter-IDS repose en grande partie sur la méthodologie adoptée pour orchestrer la composition des services. Une approche statique, où les interactions sont définies a priori, limite la capacité de réaction aux cyberattaques complexes et évolutives, lesquelles se caractérisent par leur diversité tant en termes de provenance que de nature. Assurer un niveau de sécurité élevé dans un tel environnement exige donc des mécanismes adaptatifs, permettant aux administrateurs de réagir rapidement aux nouvelles contraintes et aux changements de politique de sécurité.

C'est dans cette optique que nous avons opté pour l'intégration des règles ECA comme mécanisme de gestion de la composition des services. Ce formalisme offre un cadre de modélisation flexible et dynamique, permettant une évolution en temps réel des interactions entre services de détection d'intrusion. Contrairement aux approches traditionnelles de composition, les règles ECA permettent une reconfiguration instantanée sans nécessiter d'interruption des processus en cours, ce qui constitue un atout majeur dans le domaine de la cybersécurité [22].

Dans une perspective systémique, la coopération inter-IDS peut être assimilée à une orchestration de processus métier, où chaque service web joue un rôle spécifique dans le déroulement du processus global. La composition de ces services repose traditionnellement sur des langages dédiés tels que BPEL (Business ProcessExecutionLanguage), qui dérive des langages WSFL et XLANG [70]. BPEL permet de structurer rigoureusement l'exécution des processus métier et assure une compatibilité avec les protocoles standards des services web (SOAP, WSDL, UDDI). Toutefois, malgré sa puissance descriptive, ce langage présente plusieurs limitations qui compromettent son adaptabilité dans un cadre de coopération inter-IDS.

D'une part, sa syntaxe basée sur XML complexifie la lisibilité et la compréhension du code, rendant sa mise en œuvre laborieuse. D'autre part, son caractère impératif contraint les concepteurs à spécifier rigoureusement chaque scénario d'exécution dès la phase de modélisation, ce qui limite drastiquement la capacité d'adaptation en cas de modification des conditions opérationnelles [71]. De plus, BPEL et d'autres langages similaires, tels que WSCL et BPML, imposent une structuration figée des processus, empêchant toute évolution dynamique sans suspension du système en cours d'exécution.Dans un environnement de détection d'intrusion, où les menaces sont imprévisibles et où les attaques peuvent survenir sous des formes inédites, cette rigidité constitue un frein majeur à l'efficacité de la coopération inter-IDS. Il est donc impératif d'adopter un modèle plus souple, capable d'intégrer des changements en temps réel sans altérer la continuité opérationnelle du système.

Les langages déclaratifs, et en particulier les règles ECA, constituent une alternative particulièrement adaptée à cette problématique. Grâce à leur capacité à ajuster dynamiquement les processus en cours d'exécution, ils permettent de modifier, ajouter ou supprimer des règles sans perturber les instances actives du processus de coopération [52]. Cette flexibilité est corroborée par plusieurs travaux de recherche, notamment [52], qui soulignent que les règles ECA offrent une adaptabilité accrue grâce à leur facilité de modification et de maintenance, s'ajustant ainsi aisément aux évolutions des réglementations et politiques de sécurité.

D'après [54], les règles ECA permettent d'assurer un contrôle flexible du flux d'exécution en se basant sur des événements spécifiques. En outre, ces règles sont faciles à maintenir et favorisent l'intégration de contraintes et de déviations dans les processus de coopération [53]. L'étude menée par [72] met en avant un aspect particulièrement crucial : la gestion des exceptions [73]. En effet, la résilience d'un processus de détection d'intrusion

repose largement sur la capacité à anticiper et à gérer les anomalies survenant en cours d'exécution [74]. La prise en compte des exceptions est un défi majeur, car elle implique la modélisation de situations imprévues pouvant survenir à tout moment.Puisque les exceptions peuvent être représentées sous forme d'événements, les règles ECA offrent une solution native pour leur prise en charge, ce qui simplifie considérablement la gestion des erreurs opérationnelles [72]. Dans le domaine de la cybersécurité, cette fonctionnalité est primordiale, car l'indisponibilité d'un service IDS compromet la validité du schéma de coopération dans son ensemble [75]. Grâce aux propriétés des règles ECA, il devient possible de redéfinir dynamiquement la structure de coopération pour pallier ces lacunes et garantir une continuité opérationnelle optimale.Ainsi, au regard de ces différentes considérations, l'adoption des règles ECA constitue une réponse appropriée au défi de la flexibilité dans la détection d'intrusion coopérative. Leur capacité à orchestrer une adaptation dynamique et non intrusive en fait un levier stratégique pour la mise en place de mécanismes de coopération inter-IDS efficaces et résilients face aux menaces en constante évolution.

4. Architecture de CIIDS-SOA

L'architecture conceptuelle de notre proposition, illustrée dans la figure 17, repose sur le cadre architectural standard établi par IDWG [60]. Cette architecture modulaire est conçue pour optimiser l'efficacité des systèmes de détection d'intrusion (IDS) en améliorant l'interopérabilité et la coordination entre les différents composants. Elle se compose de quatre éléments fondamentaux, chacun remplissant un rôle stratégique dans le processus de détection et de gestion des menaces :

1. Sonde (Probe)

La sonde constitue le premier maillon du processus de détection. Il s'agit d'un module chargé de l'acquisition et de l'extraction des données provenant de diverses sources (hôte ou réseau). Ce composant surveille en permanence les activités des systèmes supervisés et identifie les comportements susceptibles d'indiquer une intrusion. Toute activité détectée comme potentiellement malveillante est convertie en un événement et transmise à l'analyseur pour une investigation approfondie. Un événement correspond donc à une activité capturée par la sonde, qui, après analyse, peut éventuellement être transformée en une alerte si elle est reconnue comme étant anormale ou suspecte.

2. **Analyseur**

L'analyseur joue un rôle central dans l'identification des intrusions. Sa mission consiste à examiner

minutieusement les événements transmis par la sonde en les confrontant à une base de connaissances, qui comprend à la fois des signatures d'attaques connues et des profils comportementaux permettant de détecter des anomalies. Dès qu'une activité suspecte est confirmée comme une tentative d'intrusion, l'analyseur génère une alerte à destination du gestionnaire (Manager). Une alerte est donc un signal émis par l'analyseur informant l'administrateur de la détection d'une menace. Ce mécanisme permet de déclencher des mesures de remédiation appropriées en temps réel.

3. Mise à jour (Update Module)

La performance et la précision de l'analyseur sont directement corrélées à la qualité de la base de connaissances sur laquelle il repose. Un système de détection basé sur des informations obsolètes peut engendrer des taux élevés de faux positifs et de faux négatifs, compromettant ainsi la fiabilité de la surveillance. Pour pallier cette problématique, l'élément de mise à jour est chargé d'actualiser continuellement la base de connaissances, en intégrant de nouvelles signatures d'attaques, en affinant les modèles comportementaux et en s'adaptant aux évolutions des techniques d'intrusion.

4. Manager (Gestionnaire des alertes et des réponses)

Le manager est le nœud décisionnel de l'architecture. Il assure plusieurs fonctions cruciales :

- La transmission des notifications aux responsables de la sécurité afin de les alerter en cas d'intrusion avérée.
- La gestion centralisée des autres composants de l'IDS, en supervisant le fonctionnement des sondes et des analyseurs.
- La réaction proactive face aux menaces, en mettant en œuvre des contre-mesures appropriées. Par exemple, en cas de détection d'une tentative d'intrusion, le manager peut ordonner le blocage de l'adresse IP suspecte, limitant ainsi les risques de compromission du système.

Rôle de l'Administrateur de Sécurité :

L'administrateur de sécurité est le pivot humain du système. Il est chargé de :

- Définir et mettre en place la politique de sécurité de l'organisation.
- Déployer et configurer les IDS en fonction des besoins spécifiques du réseau.
- Surveiller et ajuster les paramètres du système pour maximiser son efficacité.

Grâce à cette architecture modulaire et évolutive, CIIDS-SOA assure une détection d'intrusion optimisée, tout en garantissant une interopérabilité avancée entre ses différentes composantes.

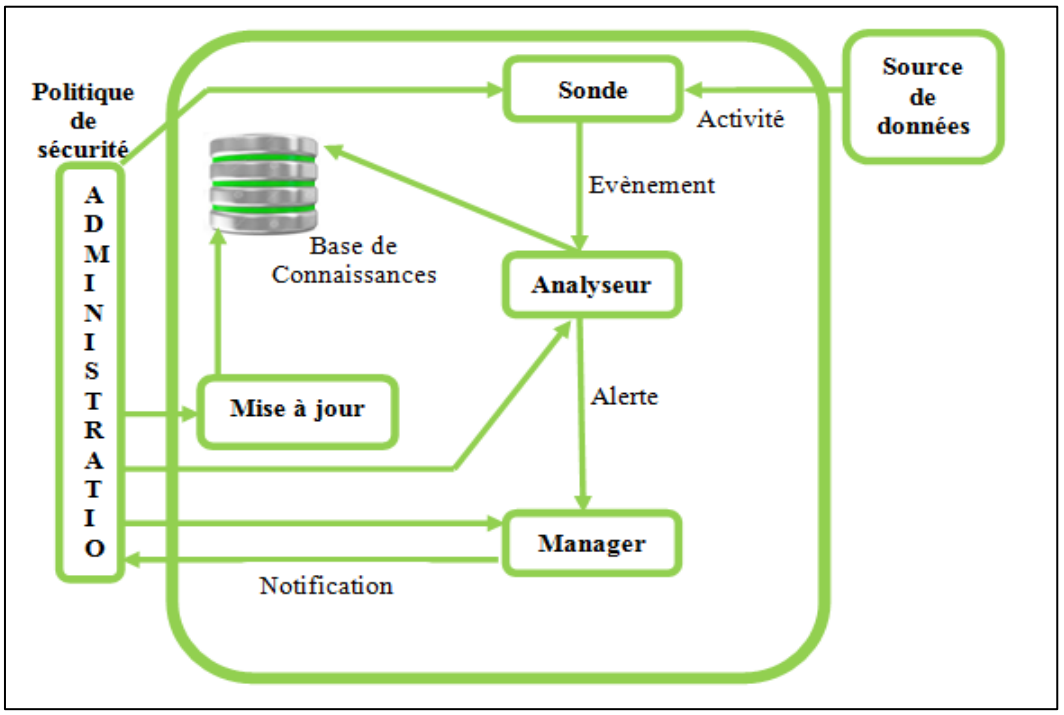

Figure 17 : Architecture de CIIDS-SOA.

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

5. Structuration CIIDS-SOA via Services Web et Règles ECA

5.1. Organisation de CIIDS-SOA basée sur l'Architecture SOA

L'architecture orientée services (SOA) se distingue par sa capacité à offrir une modularité accrue en facilitant la décomposition fonctionnelle**, le** découplage des composants et la réutilisation des services existants [67]. Ces caractéristiques fondamentales en font une approche privilégiée pour la mise en œuvre de systèmes distribués interopérables et évolutifs. Dans ce contexte, l'adoption des services webs'impose comme une solution incontournable pour assurer une communication efficace entre les différentes entités du réseau, garantissant ainsi une interopérabilité optimale au sein d'Internet [68].

Dans notre proposition, l'architecture CIIDS-SOA repose sur une organisation rigoureusement structurée exploitant pleinement les principes de la SOA. Dans cette perspective, l'élément central d'une architecture orientée services est le service lui-même. Construire une application selon cette approche implique donc la définition et l'implémentation de services autonomes et interconnectés. En examinant l'architecture d'un système de détection d'intrusion (IDS)**,** il est possible de décomposer le processus de

détection des intrusions en cinqservices principaux, chacun étant responsable d'un ensemble de tâches spécifiques et indépendantes. Ces services sont :

- Service de Capture**,**
- Service de Détection**,**
- Service de Gestion**,**
- Service d'Administration**,**
- Service de Mise à Jour**.**

Par ailleurs, le service de gestion est subdivisé en trois sous-services complémentaires :

- Service d'Alarme**,**
- Service de Réaction**,**
- Service de Pilotage**.**

5.1.1. Définition des Services Web dans CIIDS-SOA

Étant donné que notre modèle repose sur la technologie des services web, les divers services de CIIDS-SOA sont implémentés sous forme de services web interopérables, dont les rôles et interactions sont définis comme suit :

- Service de Capture : Ce module constitue l'interface de collecte des événements générés par les systèmes surveillés. Il extrait et formate les données en structures XML normalisées**,** facilitant ainsi leur analyse par le service de détection.
- Service de Détection : Il analyse les données structurées transmises par le service de capture et les confronte à une base de connaissances comprenant des signatures d'attaques connues et des modèles comportementaux. Lorsqu'une activité suspecte est détectée, ce service génère une alerte destinée au service de gestion.
- Service de Gestion : Ce service orchestre les réponses aux menaces identifiées. Il est subdivisé en trois modules fonctionnels :
    - Service d'Alarme : Génère des notifications en cas de détection d'une intrusion avérée.
    - Service de Réaction : Implémente des contre-mesures adaptées, telles que le blocage d'adresses IP malveillantes ou la **r**econfiguration des pare-feux**.**

- Service de Pilotage : Assure la coordination entre les divers composants de l'IDS et transmet les informations aux administrateurs du système.

- Service d'Administration : Il supervise la cohérence de l'ensemble des services web IDS et veille à la conformité de leur intégration au sein de l'architecture globale.
- Service de Mise à Jour : Il garantit l'actualisation continue des bases de connaissances utilisées par les IDS coopérants, assurant ainsi une détection toujours pertinente des menaces émergentes.

5.2. Vue Architecturale de CIIDS-SOA Basée sur les Services Web

5.2.1. Les Couches Fonctionnelles de CIIDS-SOA

L'architecture proposée s'articule autour de quatre couches fondamentales, chacune remplissant une fonction stratégique dans la gestion des IDS et l'orchestration des interactions entre services.

- Couche IDS : Cette couche représente l'implantation physique et logique des IDS déployés au sein du réseau. Elle inclut la configuration des capteurs, la définition des politiques de surveillance et l'intégration des IDS hétérogènes dans un cadre de coopération structuré.

- Couche Services : Dans cette couche, chaque fonctionnalité offerte par un IDS est encapsulée sous forme de service web. Elle définit l'ensemble des étapes nécessaires à l'implémentation d'un service web sécurisé, notamment : L'implémentation et le déploiement du service, la description normalisée en WSDL, l'enregistrement du service dans un registre UDDI, les méthodes d'invocation des services web, les interactions entre les services IDS et les autres composants du système.

Chaque service web peut ainsi représenter une fonction spécifique d'un IDS, voire un IDS entier, permettant ainsi l'intégration transparente d'IDS préexistants dans l'architecture CIIDS-SOA.

- Couche Composition des Services : La coopération inter-IDS repose sur un paradigme de composition des services, où la collaboration dynamique entre services assure une adaptation continue aux évolutions des menaces. Cette couche gère : l'assemblage des services IDS sélectionnés, la coordination des interactions entre services, l'orchestration des processus de détection et de réponse aux attaques. Afin de garantir une coopération flexible, réactive et facilement maintenable, les règles événement-condition-action (ECA) ont été choisies comme mécanisme de composition dynamique. Ces règles permettent d'ajuster les comportements

des services en temps réel, sans nécessiter de redémarrage du système, offrant ainsi une réactivité accrue face aux menaces émergentes.

- Couche Administration L'administration de l'architecture CIIDS-SOA constitue un élément clé de la sécurité organisationnelle. Cette couche regroupe les fonctions stratégiques associées à : la définition et l'application de la politique de sécurité, le déploiement et la configuration des IDS, la sélection des services pertinents pour la coopération inter-IDS, l'élaboration et l'adaptation des scénarios de coopération en fonction du contexte sécuritaire. Grâce à cette architecture modulaire et hautement interopérable, CIIDS-SOA constitue une approche innovante permettant de renforcer la détection d'intrusion coopérative tout en optimisant l'efficacité opérationnelle des IDS distribués.

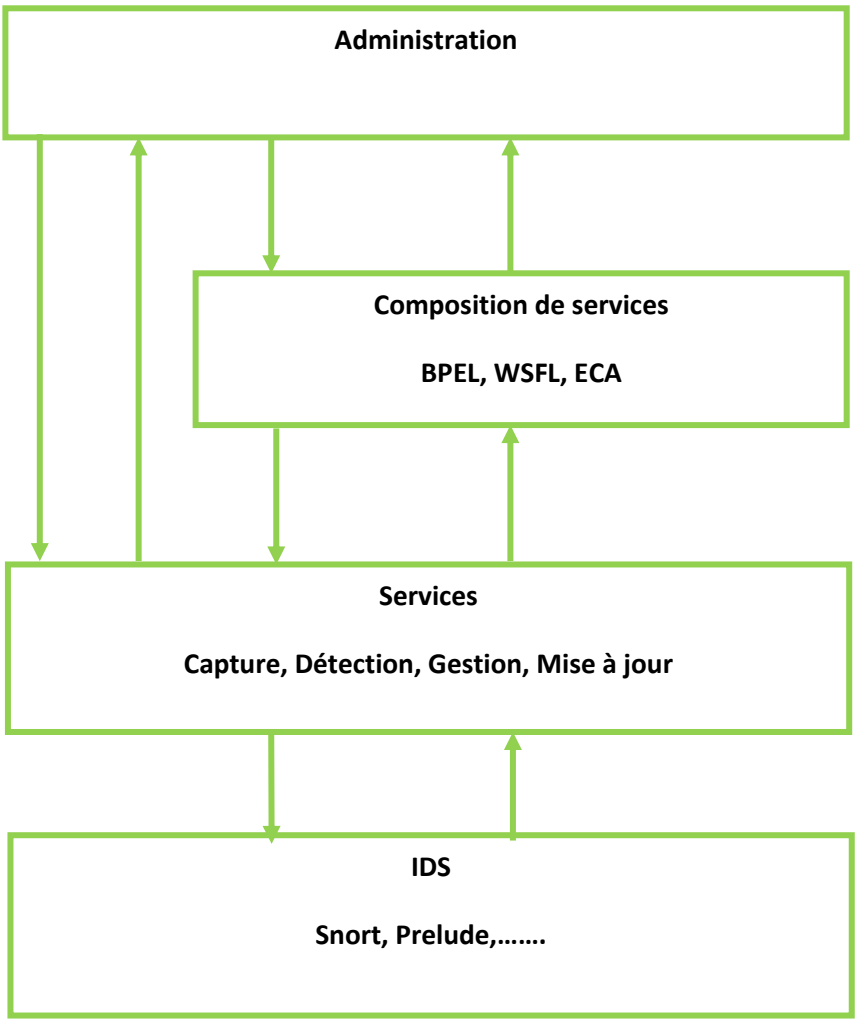

Figure18 : Vue Architecturale de CIIDS-SOA à Base Services Web

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

6. Implémentation

6.1. Configuration matérielle et logicielle

Cette section décrit l'environnement technique dans lequel notre approche a été conçue et expérimentée. L'évaluation de la solution a été menée sur deux systèmes de détection d'intrusion hétérogènes, à savoir Snort et Prelude. Le choix de ces deux IDS repose sur plusieurs considérations. Tout d'abord, ils sont tous deux open source, garantissant ainsi un accès libre et une flexibilité accrue en matière de personnalisation et d'intégration. Ensuite, ils bénéficient d'une large adoption par la communauté de la cybersécurité, ce qui facilite leur déploiement et leur interopérabilité avec d'autres solutions existantes. Snort, en particulier, est reconnu pour son efficacité et sa disponibilité sous licence GNU General Public, faisant de lui l'un des IDS les plus largement adoptés à travers le monde [78]. Il est considéré comme la référence parmi les IDS open source, avec des millions de téléchargements et une base d'utilisateurs dépassant les 250 000 inscrits [77].

Afin d'introduire une hétérogénéité dans notre expérimentation, nous avons opté pour un déploiement différencié des IDS : Snort a été installé sur un environnement Windows 7, tandis que Prelude a été configuré sous Linux Debian. Le développement des services associés à Snort a été réalisé en langage C++ à l'aide de l'environnement C++BuilderXE3, tandis que ceux liés à Prelude ont été implémentés en Java en utilisant NetBeans comme environnement de développement intégré (IDE).

6.2. Architecture implémentée

L'implémentation de notre architecture repose sur plusieurs étapes essentielles, visant à assurer la cohérence et l'efficacité du cadre de coopération inter-IDS. Ces étapes incluent :

- Le développement des différents services dédiés à l'intégration et à l'exploitation des IDS Snort et Prelude.
- L'implémentation des modules internes des deux IDS, assurant leur fonctionnement autonome et leur interaction au sein du système global.
- La mise en place du module de composition des services, garantissant l'orchestration dynamique et l'adaptation des interactions entre les différents IDS.

La figure suivante illustre l'architecture CIIDS-SOA telle qu'elle a été implémentée.

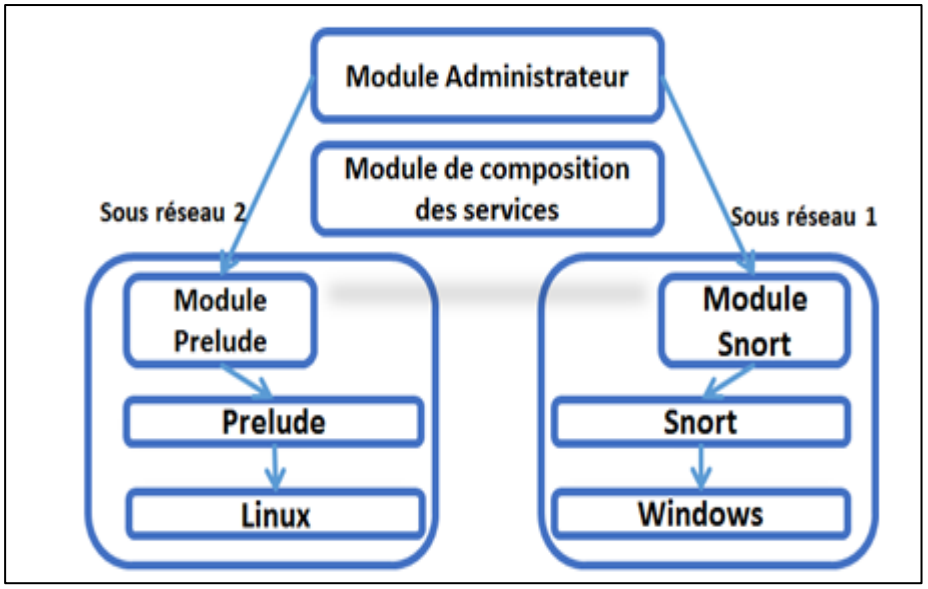

Figure 19 : Architecture implémentée de CIIDS-SOA

Source : Elaboré par le chercheur sur la base de connaissances académiques et d'analyses approfondies**.**

6.3. Module Snort

La figure suivante présente le module.

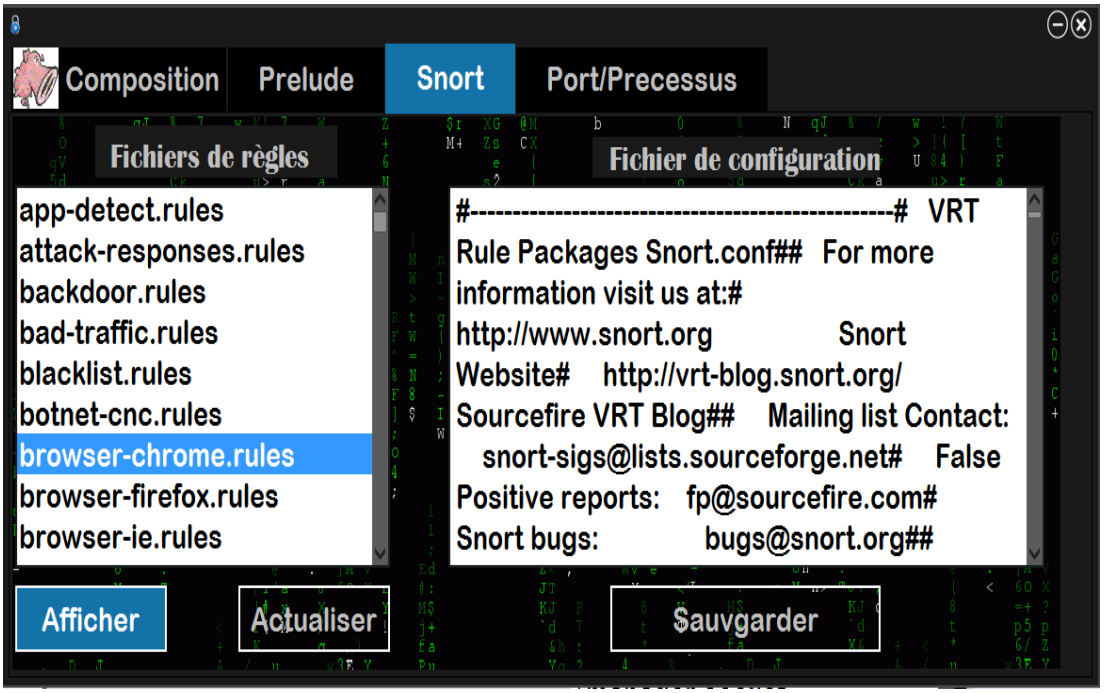

Figure 20 : Module Snort

Ce module nous permet de :

- Accéder aux fichiers de règles, sur la base desquels les données sont analysées et les intrusions détectées ; ces règles peuvent être modifiées en fonction de la politique de sécurité.

- Afficher et modifier le fichier de configuration de l'IDS Snort.

- Afficher la capture de l'IDS Snort.

6.4. Module Prelude

La figure suivante présente le module Prelude

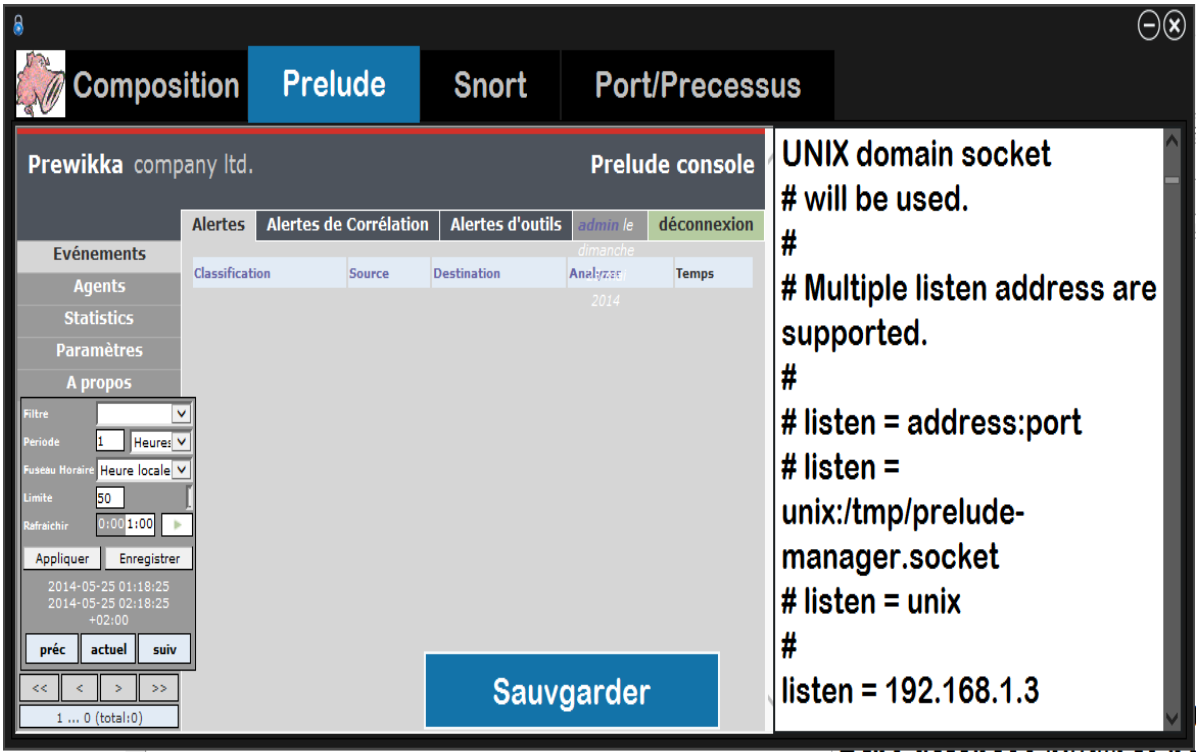

Figure 21 : Module Prelude

Ce module contient les services web pour l'IDS Prelude. Il nous permet de :

- Utiliser l'interface graphique de Prelude (Prewikka).
- Modifier et visualiser le fichier de configuration de Prelude.

6.5. Module de composition des services

À travers le module de composition des services, l'administrateur peut établir et actualiser les différents scénarios de coopération des IDS en sélectionnant les services appropriés. Nous avons défini un scénario de coopération entre les deux IDS, Snort et

Prelude, sur la base des règles ECA.

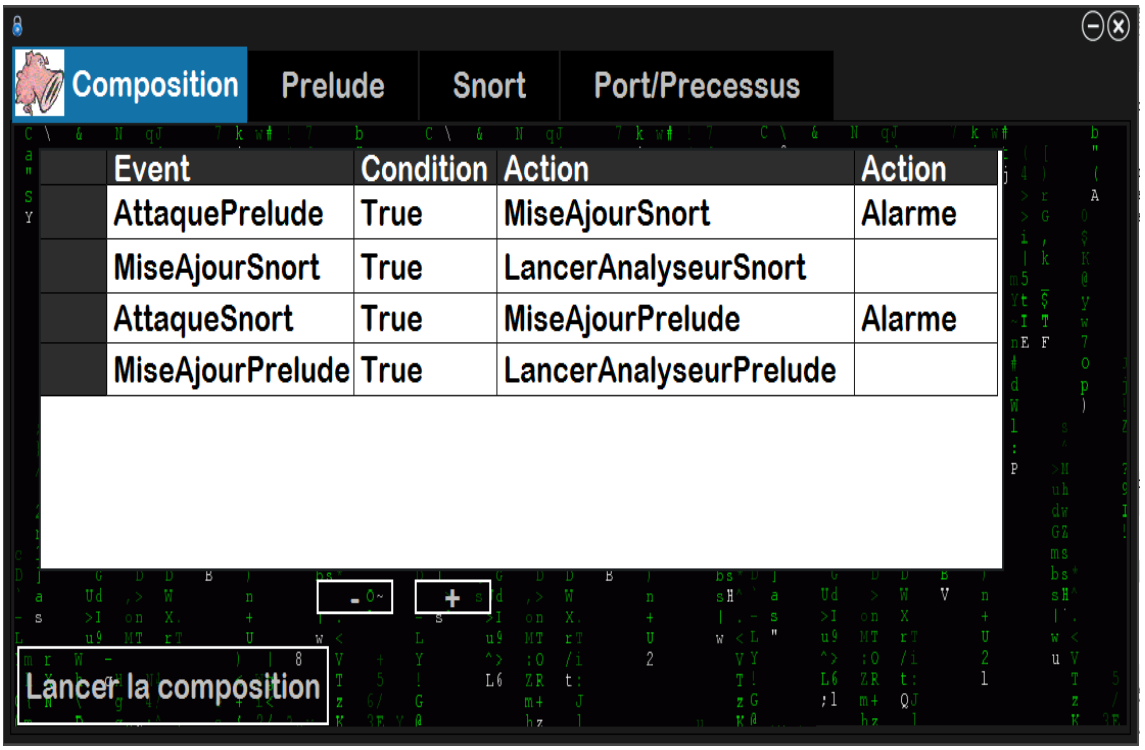

Figure 22 : Module de composition des services.

7. Test et résultats

Afin de tester notre solution, nous avons défini et exécuté un scénario de coopération entre l'IDS Snort et Prelude. Ce scénario suit la séquence d'étapes suivante :

1. Initialement, nous avons simulé une attaque de type scan sur le sous-réseau 1 protégé par Snort.

2. Snort n'a lancé aucune alarme. Il n'a pas pu détecter cette attaque, qui est ainsi passée inaperçue.

3. Sur le sous-réseau 2, protégé par Prelude, nous avons simulé la même attaque.

4. Prelude a détecté l'attaque et a généré une alarme.

5. Afin que Snort puisse détecter cette attaque ou des attaques similaires à l'avenir, nous avons procédé à une mise à jour systématique de sa base de règles à partir de l'IDS Prelude. À cet effet, nous avons établi, en fonction des règles ECA, le schéma de coopération suivant :

Événement : Attaque détectée par Prelude

Condition : Vraie

Action 1 : Mise à jour des règles de Snort

Action 2 : Génération d'une alarme

Et vice versa pour la mise à jour de la base de règles de Prelude.

6. Nous avons simulé une deuxième fois la même attaque sur le sous-réseau 1 protégé par Snort.

7. Snort a cette fois détecté l'attaque et généré une alarme.

L'alarme générée par Snort signifie qu'il a correctement détecté l'attaque cette fois-ci, démontrant ainsi que sa base de règles a bien été mise à jour.

Le résultat obtenu nous permet d'affirmer que l'architecture orientée services utilisée dans notre modèle favorise la communication et la coopération entre des IDS hétérogènes. Elle a permis à l'IDS Snort de réutiliser les fonctionnalités de l'IDS Prelude. L'intégration de la composition dynamique avec les règles ECA nous a offert la possibilité de construire le scénario adéquat avec souplesse et simplicité. Toutefois, il est essentiel de simuler plusieurs types d'attaques, notamment les attaques distribuées, afin d'évaluer plus en profondeur l'efficacité du modèle.

8. Conclusion

À travers ce chapitre, nous avons constaté que l'utilisation des règles ECA pour l'établissement souple des scénarios de coopération s'avère très prometteuse, en raison de ses avantages en termes de flexibilité et de réactivité. Nous avons ainsi présenté notre modèle visant à améliorer la flexibilité de la détection d'intrusion coopérative, la démarche suivie ainsi que les résultats obtenus.

Comme mentionné précédemment, nous avons utilisé les IDS Snort et Prelude sur des plateformes distinctes. Nous avons implémenté les modules de chaque IDS et défini un scénario de coopération entre eux.

L'objectif était d'assurer une coopération efficace entre des IDS hétérogènes en surmontant les difficultés liées à l'administration du réseau et en minimisant le temps nécessaire à l'établissement des scénarios de coopération. Les résultats obtenus démontrent une réutilisation optimisée des fonctionnalités entre les IDS, tout en garantissant une plus grande flexibilité et adaptabilité.

Conclusion Générale

Conclusion générale

Dans un monde où les réseaux informatiques constituent l'épine dorsale des communications et des échanges numériques, la sécurisation des infrastructures devient un enjeu prioritaire. L'évolution exponentielle des technologies de l'information a engendré une prolifération des cyberattaques, de plus en plus sophistiquées et complexes, mettant en péril l'intégrité, la confidentialité et la disponibilité des données. Face à cette menace croissante, les mécanismes de défense traditionnels, notamment les systèmes de détection d'intrusions (IDS), se révèlent souvent insuffisants en raison de leur rigidité et de leur incapacité à s'adapter aux environnements dynamiques et distribués.

Dans ce contexte, notre travail s'est inscrit dans une perspective d'amélioration de la détection d'intrusions coopérative, en exploitant les potentialités de l'architecture orientée services (SOA). Nous avons procédé à une analyse approfondie des IDS existants, mettant en exergue leurs avantages et leurs limites structurelles. Il est apparu clairement que l'adoption d'une approche unique de détection ne permet pas de couvrir l'ensemble des menaces émergentes, et que la mise en place d'une coopération inter-IDS est devenue une nécessité incontournable. Toutefois, cette coopération se heurte à une hétérogénéité des technologies employées, des protocoles de communication et des formats d'échange de données.

Pour répondre à cette problématique, nous avons exploré les mécanismes de standardisation et d'interopérabilité proposés par la communauté scientifique, et nous avons analysé les travaux antérieurs visant à intégrer SOA dans la détection d'intrusions coopérative. Ces études ont démontré que SOA constitue une approche prometteuse, offrant une modularité accrue, un couplage lâche entre les composants et une flexibilité adaptée aux environnements complexes. Cependant, nous avons relevé que les solutions existantes souffrent encore de limitations en termes d'adaptabilité, de scalabilité et de réactivité aux attaques en temps réel.

C'est dans cette optique que nous avons proposé une nouvelle approche basée sur SOA, visant à surmonter les défis liés à l'interopérabilité entre les IDS et à améliorer leur capacité de réaction face aux menaces émergentes. Notre contribution repose sur l'intégration d'une composition dynamique des services de détection, rendue possible grâce à l'exploitation des règles événement-condition-action (ECA). Ce mécanisme nous a permis de concevoir une architecture plus souple et adaptative, où les IDS peuvent collaborer de manière fluide et autonome, sans dépendre d'une configuration statique rigide.

L'expérimentation de notre approche a été réalisée sur deux IDS hétérogènes : SNORT et PRELUDE. Nous avons procédé au développement des services web nécessaires, assurant l'interopérabilité entre ces systèmes, puis à leur intégration au sein d'une architecture SOA. Un scénario de détection coopérative a été mis en place afin de valider l'efficacité de notre modèle, démontrant une amélioration notable dans la gestion et la corrélation des alertes.

Notre travail était sanctionné par une communication dans une conférence [79].

Bien que notre approche ait permis d'améliorer la coopération inter-IDS et d'accroître la réactivité des mécanismes de détection, plusieurs perspectives restent ouvertes pour approfondir ce travail et le rendre encore plus robuste face aux défis de la cybersécurité. Parmi les pistes envisageables, nous pouvons citer :

- L'intégration d'un module de vérification de la composition des services, garantissant une cohérence stricte entre les différentes étapes du processus de détection coopérative.
- Le développement d'un module avancé de corrélation d'alertes, permettant d'identifier des schémas d'attaques distribuées et multistages avec une précision accrue.
- L'exploration des techniques d'intelligence artificielle et de machine learning, afin d'optimiser l'adaptabilité du système et d'anticiper les menaces émergentes.
- L'étude approfondie de l'aspect sécuritaire des services web, en intégrant des mécanismes avancés de chiffrement et d'authentification pour renforcer la protection des échanges entre IDS.

Ces évolutions constituent des leviers stratégiques pour perfectionner les infrastructures de cybersécurité basées sur SOA, en les rendant plus dynamiques, plus intelligentes et plus résilientes face aux cybermenaces de demain.

Références Bibliographiques

[1] International Standards Organization. Information Processing Systems - OSI –Reference Model - Part 2: Security Architecture. Technical report 7498-2, 1989.

[2] J. BRIFFAUT. Formalisation et garantie de propriétés de sécurité système : application à la détection d'intrusions. Thèse doctorat. université d'Orleans. 2007.

[3] J. P. Anderson. Computer security threat monitoring and surveillance. Technical report, James P. Anderson Company. Fort Washington. 1980.

[4] A. Lazarevic, V. Kumar et J. Srivastava. Intrusion detection: a survey, Managing Cyber Threats, approaches, and challenges. Springer. 2005.

[5] A. Singhal. Intrusion Detection Systems. in Data Warehousing and Data Mining Techniques for Cyber Security. Springer. 2007.

[6] P. Kazienko et P.Dorosz. Intrusion Detection Systems (IDS) Part I- (network intrusion; attack symptoms; IDS tasks; and IDS architecture). 2004

http://www.windowsecurity.com/pages/article_p.asp?id=1147.

[7] A. Abraham et R. Jain. Soft Computing Models for Network Intrusion Detection Systems. In : K. Halgamuge S, Wang L (éd.)Springer Berlin / Heidelberg. 2005.

[8] J. Aycock. Computer Viruses and Malware. Springer. 2006.

[9] I. S. Winkler et B. Dealy. Information security technology?...Dont rely on it. A case study in social engineering. Proceedings of the Ninth Usenix Security Symposium.1995.

[10] E. Spafford, et D. Zamboni, Intrusion Detection Using Autonomous Agents, Computer Networks, vol. 34, pp. 547-570, 2000.

[11] K. Scarfone et P. Mell. Guide to intrusion detection and prevention systems (idps). Technical report, National Institute of Standards and Technology. Special Publication 800-94. USA. 2007.

[12] NIST. Intrusion detection Systems. NIST Computer Science Special reports. SP 800-31. November 2001.

[13] N. R. Peddisetty. State-of-the-art Intrusion Detection Technologies, Challenges, and Evaluation. Linkoping. Feb 2005.

[14] H. Debar, M. Dacier et A. Wespi. Towards a Taxonomy of Intrusion Detection Systems. Computer Networks. vol. 31, 8, pp. 805-822. 1999.

[15] P.A. Porras et A. Valdes. Live Traffic Analysis of TCP/IP Gateways. In Proceedings of the ISOC Symposium on Network and Distributed System Security (NDSS'98), San Diego. CA. March 1998.

[16] F.Sabahi et A.Movaghar. Intrusion Detection: A Survey. The Third International Conference on Systems and Networks Communications. IEEE, 2008.

[17] Zwicky E. D., Cooper S., Chapman D. B. Building Internet firewalls (2nd ed.). Sebastopol, CA, USA : O'Reilly& Associates, Inc., 2000.

[18] A. Ghorbani, W. LuetM. Tavallaee. Network Intrusion Detection and Prevention. in Advances in Information Security,book. Springer. 2010.

[19] F.Majorczyk. Détection d'intrusions comportementale par diversification de COTS : application au cas des serveurs web. Thèse de doctorat, université de Rennes 1. Décembre 2008.

[20] J. Zimmermann et L. Mé. Les systèmes de détection d'intrusions: principes algorithmiques. Multi-System & Internet Security Cookbook, 2002.

[21] L.Me et V.Alanou. Détection d'intrusion dans un système informatique : méthodes et outils. TSI. Technique et science informatiques. 1996. Vol. 15, n°4, p. 429-450.

[22] M.Boukhebouze . Gestion de changement et vérification formelle de processus métier : une approche orientée règle. Thèse de doctorat. L'institut national des sciences appliquées de Lyon. 2010.

[23] M. Roesch. Snort- Lightweight Intrusion Detection for Networks. In LISA"99: Proceedings of the 13th USENIX conference on System administration, p. 229-238. Berkeley, CA, USA. 1999.

[24] K. J. Cox, C. Gerg. Managing Security with Snort and IDS Tools. Publisher: O'Reilly - ISBN : 0-596-00661-6, 2004.

[25] A. R. Baker, B. Caswell et M. Poor. Snort 2.1 Intrusion Detection Second Edition. Shroff Publishers & Distributors PVT. LTD. 2004.

[26] L. Me et V. Alanou. Détection d'intrusion dans un système informatique : méthodes et outils. TSI. Technique et science informatiques. Vol. 15, n°4, p. 429-450. 1996.

[27] M. MacKenzie, K. Laskey, F. McCabe, P. Brown, et R. Metz. Reference Model for Service-Oriented Architecture 1.0. Technical Report wd-soa-rmcdl, OASIS.

[28] OASIS. Reference architecture for service oriented architecture 1.0. April 2008. http ://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf.

[29] Gartner. Hype Cycle for Web Services. http://www.gartner.com/, 2007.

[30] Y. Charif. Chorégraphie dynamique de services basée sur la coordination d'agents introspectifs. Thèse de doctorat. université pierre et marie curie. Paris VI.2007.

[31] The Workflow Management Coalition. Workflow Management Coalition Terminology & Glossary. Rapport numéro WFMC-TC-1011 Issue 3.0. Février 1999.

[32] S. Dustdar et W. Schreiner. A Survey on Web services Composition. International Journal of Web and Grid Services, 1. 2005.

[33] T. Osman, D. Thakker et D. Al-Dabass. Bridging the Gap between Workflow and Semantic-based Web services Composition. In Proc. of the Web Service Composition Workshop WSCOMPS05, 2005.

[34] A. Barros, M. Dumas, and P. Oaks. A Critical Overview of the Web Services Choreography Description Language (WS-CDL). In Proc. of the Business Process Trends (BPTrends), 2005.

[35] C. Peltz. Web Services Orchestration and Choreography. Computer, 26(10) :46–52, 2003.

[36] IBM, Microsoft, SAP, Siebel Systems. Business Process Execution Language for Web Services Version 1.1. Technical report, 2003.

[37] J. Moon, D. Lee, C. Park, et H. Cho. ebXML BP Modeling Toolkit. Dans Proc. of the 7th International Conference on Enterprise Distributed Object Computing (EDOC), page 296. IEEE Computer Society, 2003.

[38] F. POURRAZ, Diapason une approche formelle et centrée architecture pour la composition évolutive de services Web. Thèse de Doctorat 2007.

[39] N. Kavantzas, D. Burdett, G. Ritzinger, T. Fletcher, and Y. Lafon. Web Services Description Language (WS-CDL). Technical report, W3C, 2004.

[40] The Workflow Management Coalition. Final XPDL 2.1 Specification. Dans le Rapport de spécification numéro WFMC-TC-1025-Oct-10-08-A. 2008.

[41] OASIS. Collaboration-Protocol Profile and Agreement Specification. ebXML Trading-Partners Team Version 1.0. Dans ebXML specification 2003.

[42] T. Crusson. Business Process Management : De la modélisation à l'exécution ». Dans le rapport technique (Livre Blanc) Intalio, 2003.

[43] F. Leymann. Web Services Flow Language (WSFL 1.0). Dans

http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf. 2001.

[44] S. Thatte. XLANG – Web Services For Business Process Design. Dans http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm. 2001.

[45] OASIS. Business Process Execution Language for Web Services (BPEL4WS 2.0). Dans le rapport de spécification. 2007.

[46] Intalio et BPMI. Bussiness process modeling language. dans http://www.bpmi.org/bpmi-downloads/BPML-SPEC-1.0.zip, 2002

[47] T. Andrews, F.Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic et S.Weerawarana, 2003a, Business Process Execution Language for Web Services.2003 , dans http://www.ibm.com/developerworks/library/specification/ws-bpel/.

[48] P. Chulsoon et C. Injun .Management of business process constraints using BPTrigger. Dans la revue Computers in Industry 55 (2004) 29–51. 2004.

[49] M. zur Muehlen, M. Indulska et G. Kamp . Business Process and Business Rule Modeling: A Representational Analysis. Dans 3rd International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2007), Annapolis, Maryland, USA, 15 Octobre, 2007.

[50] G. Wagner .Rule Modeling and Markup .Dans Reasoning Web, 3564 ed, N. Eisinger and J. Maluszynski, Eds. Msida, Malta: Springer, 2005, pp. 251-274.

[51] G. Knolmayer, R. End,et M. Pfahrer .Modeling Processes and Workflows by Business Rules. Dans Business Process Management, Models, Techniques, and Empirical Studies W. M. Aalst, J. Desel, and A. Oberweis, Eds. Lecture Notes In Computer Science, vol. 1806. Springer- Verlag, London. 16-29. 2000

[52] F. Bry, M. Eckert,P. L. Pătrânjan et L. Romanenko . Realizing Business Processes with ECA Rules: Benefits, Challenges, Limits. Dans 4th International Workshop, PPSWR 2006, Budva, Montenegro, June 10-11, 2006.

[53] G. Wagner, A. Giurca, et S. Lukichev .Modeling Web Services with URML. Dans proceedings of Semantics for Business Process Management Workshop, Budva, Montenegro, June 2006.

[54] A. Giurca, S. Lukichev, et G. Wagner . Modeling Web Services with URML .Dans Proceedings of SBPM2006, Budva, Montenegro (11th June 2006), June 2006.

[55] K.V.S.N Rama Rao, M. R. Patra . A Service Oriented Modeling and Analysis for Building Intrusion Detection Systems. Dans Global Trends in Computing and Communication Systems Communications in Computer and Information Science Volume 269, pp 661-670. 2012.

[56] S. Staniford-Chen, B. Tung, Ph. A. Porras, C. Kahn, D. Schnackenberg, R. Feiertag, et M.Stillman. The common intrusion detection framework-data formats, internet draft. Technical report. March 1998.

[57] L. Me, Z. Marrakchi, C. Michel, H. Debar, et F. Cuppens. La detection d'intrusion : les outils doivent cooperer. Dans REE Journal, pp. 50-55, No 5. May, 2001.

[58] A. Bosin, N. Dessi, et B. Pes. A Service Based Approach to a New Generation of Intrusion Detection Systems- on Web Services. ECOWS '08. IEEE Sixth European Conference. 2008.

[59] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, et E. Stoner. State of the practice of intrusion detection technologies. Technical report, CarnegieMellon Software Engineering Institute, January 2000.

[60] Intrusion Detection Exchange Format (idwg). Dans : http://datatracker.ietf.org/wg/idwg/charter.

[61] Intrusion Detection Interoperability and Standardization, 2002. Dans : http://cs.uccs.edu/~chow/pub/master/sjelinek/doc/research/idmef.pdf

[62] J.E. Brandao, P.M. Mafra, et J.S. Fraga. A New Approach for IDS Composition. Dans Proceedings of the IEEE International Conference on Communications (ICC 2006), IEEE, 2006.

[63] J.E. Brandao, P.M. Mafra, J.S. Fraga, et R.R. Obelheiro. A WS-Based Infrastructure for Integrating Intrusion Detection Systems in Large-Scale Environments. LNCS, vol. 4275, Springer-Verlag, 2006.

[64] L. Fagundes et L.P. Gaspary.Breaking the Barriers between Security Mechanisms through the Composition of Web Services: Towards a Solution for the Detection of Multistage Distributed Attacks. Computers and Communications, IEEE. 2009.

[65] A.R.Roozbahani, R. Nassiri, et L.Shabgahi. Service Oriented Approach to Improve the Power of Snorts. Computer and Electrical Engineering. 2009.

[66] W. E. L.Costa, D. Lopes, Z. Abdelouahab, et B. Froz. Network Intrusion Detection System Based on SOA (NIDS-SOA): Enhancing Interoperability Between IDS- Innovations and Advances in Computer, Information, Systems Sciences, and Engineering . Lecture Notes in Electrical Engineering Volume 152, 2013.

[67] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, et D. F. Ferguson. Web Services Platform Architecture : SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WSReliable Messaging and More. Prentice Hall PTR, Upper Saddle River, NJ, USA. 2005.

[68] M. Hirzalla, J. Cleland-Huang, et A. Arsanjani. A Metrics Suite for Evaluating Flexibility and Complexity in Service Oriented Architectures. In the 6th International Conference on Service Oriented Computing, ICSOC Workshops, pp. 41-52. 2008.

[69] M.Rouached. RESTful Web Services for High Speed Intrusion Detection Systems. IEEE, 2013.

[70] S. Leutenmayr. Selected Languages for Web Services Composition: Survey, Challenges, Outlook. Dans thèse de doctorat à Université Louiset- Maximilien de Munich. 2007.

[71] M. Boukhebouz. Gestion de changement et vérification formelle de processus métier : une approche orientée règle. Dans thèse de doctorat à l'institut national des sciences appliquées de Lyon. 2010.

[72] S. Goedertier,R.Haesen, et J. Vanthienen .EM-BrA²CE v0.1: A Vocabulary and Execution Model for Declarative Business Process Modeling. Dans rapport de recherche université de technologie Eindhoven, 2007.

[73] H. Debar, M. Dacier, et A. Wespi. "Vers une taxonomie des systèmes de détection d'intrusions." Dans Technique et Science Informatiques, vol. 18, no. 8, 1999, pp. 1065-1088.

[74] F. Cuppens et A. Miege. "Modèle de coopération pour la détection d'intrusions." Dans Journal de la Sécurité Informatique, vol. 1, no. 2, 2002, pp. 45-60.

[75] Y. Deswarte et M. Kaaniche. "Sécurité des systèmes d'information : concepts et mécanismes." Dans Revue Technique de l'Ingénieur, vol. 3, no. 5, 2004, pp. 12-25.

[76] M. Grégoire. "Les architectures orientées services (SOA) : principes et mise en œuvre." Dans Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information, vol. 13, no. 5, 2008, pp. 11-30.

[77] URL: http://www.snort.org

[78] Z. Afzal , S. Lindskog, Multipath TCP IDS Evasion and Mitigation, Information security, 18th International Conference, ISC 2015, Volume pp 265-282, August 2015.

[79] Y. Dahmani, K. Bekki et K. Bekki. Une approche de cooperation entre IDS base sur SOA. Dans Colloque sur l'optimisation et les Systèmes d'Information COSI 2015, Oran, Algérie. Juin 2015

[80] Briser les barrières entre les mécanismes de sécurité à travers la composition de services Web: vers une solution pour la détection des attaques distribuées Multistage

[81] K. Boudaoud et S. Belguith. "Sécurité des services web dans les architectures orientées services." Dans Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information, vol. 15, no. 1, 2010, pp. 59-84.