*République Algérienne Démocratique et Populaire*
*Ministère de l'enseignement supérieur et de la recherche scientifique*
*Université Larbi Tébessi - Tébessa*
*Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie*
*Département : Mathématiques et Informatique*

*Mémoire de fin d'étude*
*Pour l'obtention du diplôme de MASTER*
*Domaine : Mathématiques et Informatique*
*Filière : Informatique*
*Option : Réseau et Sécurité Informatique*

*Thème*

# Systeme de Detection d'Intrusion basée sur Les Systémes Multi-Agents

*Présenté Par :*
*BREK Bouthaina*

*Devant le jury :*

| | | | |
|---|---|---|---|
| *Mr.A.Sahraoui* | *MAA* | *Univérsité Larbi Tébessa* | *Président* |
| *Mr.Y.Menassel* | *MAA* | *Univérsité Larbi Tébessa* | *Examinateur* |
| *Mr. Ali Abdelatif Betouil* | *MCB* | *Univérsité Larbi Tébessa* | *Encadreur* |

*Date de soutenance : 23 /06/2019*

# Résumé

L'implémentation d'une capacité de détection d'intrusion efficace est un objectif difficile à atteindre, qui n'est pas résolu facilement. Cependant, nous affirmons que la technologie multi-agents et précisément un agent mobile contribue grandement à la réalisation du comportement idéal souhaité dans un système de détection d'intrusion (IDS), Pour améliorer les performances des agents mobiles, nous les rendons intelligents en utilisant les techniques de la Machine Learning. Ce mémoire traite les différentes manières d'appliquer les agents mobiles au problème de la détection et de la réponse aux intrusions. Ce mémoire n'examine pas seulement les avantages tirés de la mobilité, mais également ceux associés aux agents logiciels en général. Après avoir exploré ces avantages, nous décrivons un certain nombre de façons d'appliquer la technologie des agents mobiles pour remédier aux faiblesses des conceptions et des implémentations des IDS actuelles, et définissons les problèmes de sécurité associés.

**Les mots clés :** Detection d'intrusion, Système Multi-agents, Agent Mobile, Machine Learning

# Abstract

Implementing an effective intrusion detection capability is a difficult goal to achieve, which is not easily solved. However, we assert that multi-agent technology and precisely a mobile agent greatly contributes to achieving the desired ideal behavior in an intrusion detection system (IDS), and to improve the performance of mobile agents, we make them intelligent in using machine-learning techniques. This paper discusses different ways to apply mobile agents to the problem of intrusion detection and response. This paper examines not only the benefits derived from mobility, but also those associated with software agents in general. After exploring these benefits, we describe a number of ways to apply mobile agent technology to address weaknesses in current IDS designs and implementations, and define associated security issues.

**Keywords:** Intrusion Detection, Multi-Agent System, Mobile Agent, Machine Learning

# ملخص

يعد تحقيق قدرة فعالة لاكتشاف الاختراق هدفًا صعبًا لا يمكن تحقيقه بسهولة. ومع ذلك ، فإننا نؤكد أن التكنولوجيا متعددة الوكلاء وعلى وجه التحديد الوكيل المتنقل يساهم إلى حد كبير في تحقيق السلوك المثالي المرغوب فيه في نظام كشف التسلل (IDS) ، ولتحسين أداء الوكلاء المتنقلة، نقوم باضافة خاصية الذكاء وذلك باستخدام تقنيات التعلم الآلي. تتناول هذه الورقة طرقًا مختلفة لتطبيق وكلاء الجوال على مشكلة اكتشاف التسلل والاستجابة له. لا يبحث هذا البحث فقط على الفوائد المستمدة من التنقل ، ولكن أيضًا الفوائد التي يرتبط بها وكلاء البرامج بشكل عام. بعد استكشاف هذه الفوائد ، نصف عددًا من الطرق لتطبيق تكنولوجيا وكلاء الأجهزة المحمولة لمعالجة نقاط الضعف في تصميمات IDS الحالية وتنفيذها ، وتحديد مشكلات الأمان المرتبطة بها.

**الكلمات المفتاحية:** كشف التسلل ، نظام متعدد الوكلاء ، وكيل المحمول ، تعلم الآلة

# Remerciements

*Je tiens à remercier tout d'abord, le Dr. Ali Abdelatif Betouil, de m'avoir proposé un tel intéressant sujet, pour son encadrement avec patience, ses précieux conseils, sa disponibilité et son soutien tout au long de ce travail.*

*Je tiens à remercie également les membres du jury d'avoir accepté d'évaluer mon travail.*

*Mes remerciements à tous ceux qui m'ont aidé de près ou de loin…*

# Dédicace

*Je dédie ce travail qui n'aura jamais pu voir le jour sans les soutiens indéfectibles et sans limite de mes chers parents, ma chère sœur et mes deux frères qui ne cessent de me donner avec amour le nécessaire pour que je puisse arriver à ce que je suis aujourd'hui. Que dieux vous protège et que la réussite soit toujours à ma portée pour que je puisse vous combler de bonheur.*

## Table des matières

# Liste des figures

# Introduction Générale

**Introduction générale :**

Au début de son apparition, l'internet a été restreint seulement pour les gouvernements, les militaires et les chercheurs académiques et pour ça il n'y a pas un grand intérêt avec la sécurité de l'information et le développement des protocoles de sécurité. Avec le développement rapide et énorme des réseaux ces dernières années, les informations virtuelles sont devenues très risquées, au niveau personnelles ou professionnelles, il est donc devenu nécessaire de les protéger.

Plus que jamais, nous constatons qu'Internet modifie l'informatique telle que nous la connaissons. Les possibilités et les opportunités sont illimitées, tout comme les risques et les intrusions malveillantes et les attaques informatique. Une attaque est une menace à la sécurité de l'information qui consiste à obtenir, modifier, détruire, supprimer, implanter ou révéler des informations sans accès autoriser ou authentification. Cela arrive à la fois aux individus et aux organisations et de nos jours, le cout d'une attaque et de sa réparation peut être élevé. Il existe de nombreux types d'attaques, notamment les attaques passives, actives, les virus, les vers, les spams, les machines zombies, et les attaques par déni de service…etc., Et pour cette raison, les chercheurs s'orientent nécessairement vers la sécurité informatique, de sorte que plusieurs branches ont été créés dans cette domaine notamment Firewall, Anti-virus, VPN, IDS etc.

James Anderson [13] a été le premier à introduire le concept de systèmes de détection en 1980, mais le premier modèle a été créé par Denning Dorothy en 1987 [14]

La recherche dans ce domaine est en cours et de nombreux prototypes sont développés selon leur différents types et architectures. Et les gouvernements qui contrôlent les domaines scientifiques et de la recherche sont toujours en concurrence pour accroître la sécurité des technologies de l'information en effectuant des investissements importants dans le domaine de détection d'intrusion.

Dans ce mémoire on va proposer un modèle d'IDS basé sur les systèmes multi-agent, chaque agent est un machine learning et qui transite entre les différents nœuds du réseau.

Ce mémoire est structuré comme suit :

- Chapitre 1 : on parle en générale sur la sécurité informatique et on détail sur les systèmes de détection d'intrusion et leur différent types et architecture.
- Chapitre 2 : On parle sur les systèmes multi-agent et leurs caractéristiques et les différents types d'agent.
- Chapitre 3 : représente notre proposition et l'implémentation de notre model avec les résultats obtenus.

# *Chapitre 1 :*

## Sécurité informatique et IDS

## I. Sécurité informatique :

### 1. Introduction :

Les progrès technologiques, le développement des moyens de communication, l'ouverture du monde aux nouvelles technologies et le transfert de divers types de données sur des réseaux, les rendant vulnérables aux menaces à la sécurité émanant de personnes non autorisées ou de concurrents. La protection des informations est devenue une nécessité pour les particuliers ou les entreprises. Pour ce faire, elle utilise des techniques et des mécanismes d'authentification et un contrôle de l'accès. Des outils doivent être utilisés pour garantir la sécurité des informations, notamment Firewall, VPN, IDS…etc. Le système de détection d'intrusion (IDS) est l'un de ces outils cruciaux des opérations de défense, La détection d'intrusion consiste essentiellement à rechercher des signes d'attaque. Lorsqu'une intrusion est détectée, le système de détection d'intrusion peut prendre Les procédures nécessaire Selon son type et sa programmation par l'administrateur.

2. **La sécurité informatique** : C'est l'ensemble de technologies utilisées pour réduire les vulnérabilités du système d'information contre les attaques accidentelles ou intentionnelles et ça c'est le but d'utiliser la sécurité pour le système d'information. La sécurité informatique est caractérisée généralement par les cinq propriétés ou objectifs suivants **[1]**:

- **Disponibilité** : Lorsqu'un utilisateur du système d'information demande des informations, la ressource doit être disponible pour répondre aux personnes autorisées uniquement.
- **Confidentialité** : assurer que les informations sont cachées sur le système afin qu'elles ne puissent être lue que par les personnes autorisées.
- **Intégrité** : Garantit l'impossibilité de modifier les informations relatives au système par des individus non autorisés sans l'intervention des personnes autorisées sans avoir les informer.
- **Non-répudiation:** Est de prouver la source de données pour que les individus ne puissent pas nier leur participation à la communication.
- **L'authentification :** Est d'assurer l'identité de l'utilisateur, dans le sens que doit être garantir l'identité de chacune des parties impliquées dans la communication, aussi il faut

également assurer le contrôle d'accès aux ressources pour les individus autorisés (l'accès au compte e-mail avec une adresse et mot de passe correcte).

3. **Système d'information :**

Le système d'information est l'ensemble des moyens technologiques, organisationnels et humains permettant de collecter, stocker, traiter et distribuer de l'information entre les organisations **[2]**.

### 3.1.Sécurité des systèmes d'information :

La sécurité des systèmes d'information est atteinte lorsque les objectifs de sécurité tel que la disponibilité, l'intégrité, la confidentialité, l'authentification et la non-répudiation sont garantis d'être réalisé, et confirmer que les méthodes, techniques et outils nécessaires à la protection des ressources du système d'information sont utilisés **[3]**.

4. **Domaines de la sécurité :**

L'informatique étant l'un des fondements de l'entreprise, qui intervient également dans tous les domaines, c'est pour ça il est nécessaire de veiller à la sécurité des systèmes d'information. Selon leur domaine d'application, les moyens de la sécurité se classifient en **[4] [5]** :

- Sécurité physique
- Sécurité de l'exploitation
- Sécurité logique
- Sécurité applicative
- Sécurité des télécommunications

On retrouve aussi dans le domaine de la sécurité informatique l'usage de quelques termes qu'il faut les reconnaitre, parmi eux : **[6][7] :**

- **Vulnérabilité** : Une faute dans le système d'exploitation créée durant son développement ou c'est une faiblesse dans la sécurité des systèmes d'informations ou des réseaux, etc., qui peut être exploitée sous forme d'une menace sur la sécurité est utilisée pour pénétrer les systèmes et les réseaux.

- **Intrusion** : Action malveillante résultant d'une attaque externe qui a réussi à exploiter une vulnérabilité pour permettre à l'attaquant de contrôler le système ou le réseau.

- **Menace** : possibilités et probabilités d'attaque contre la sécurité. Une menace est définie par le processus d'attaque, par la cible et par le résultat (conséquences de la réussite d'une attaque**).**

- **Attaque**: C'est n'importe quelle action qui a le but de menacer la sécurité des informations et de nuire au moins à l'une des propriétés de la sécurité informatique (disponibilité, Confidentialité, Intégrité, L'authentification). Il s'agit d'une tentative d'intrusion, nous abordons dans ce qui suit les différents buts et classes de ces attaques (tentatives d'intrusion).

**5. Les types d'attaque informatique :**

Les systèmes informatiques utilisent différents composants, allant de l'électricité aux machines en fonctionnement, en passant par les programmes exécutés sur le système d'exploitation et utilisant le réseau. Des attaques peuvent se produire dans chaque lien vers cette chaîne, s'il existe une faille pouvant être exploitée [8]. Pour l'aspect technique, on définit que l'attaque c'est une exploitation d'une faille pour des fins illégales. Il existe cinq formes d'attaque que nous détaillons comme suit [9] :

- **L'attaque passive :** Les attaques passives sont toute action nous permettant d'analyser et de déchiffrer le trafic, de surveiller les communications et de capturer des informations d'authentification. L'attaquant utilise ce type d'attaque pour obtenir des informations ou de données facilement et à l'insu de la victime en interceptant les mots de passe, les numéros de carte de crédit et les emails.

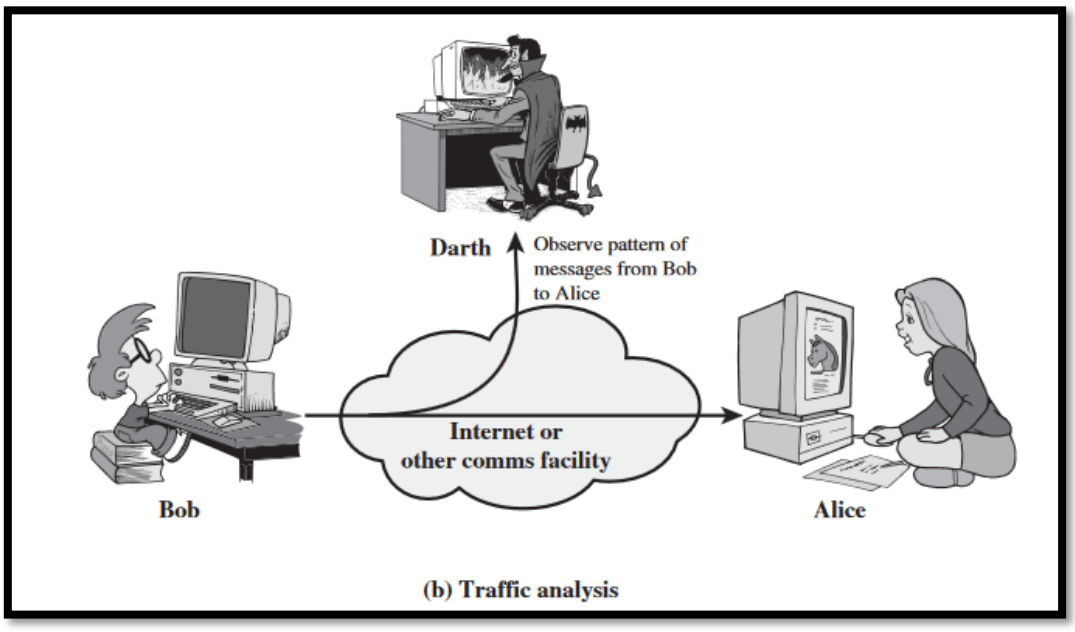



**Figure 1 :** L'attaque passive (L'analyse du trafic réseau) [11]

- **L'attaque active :** Les attaques actives incluent les tentatives visant à contourner ou casser des fonctionnalités de sécurité afin de falsifier ou de voler des informations en insérant un code malveillant dans le système d'exploitation ou le réseau, ainsi que des menaces d'attaques actives visant à détecter ou à publier des fichiers de données, à refuser le service ou à modifier les données.

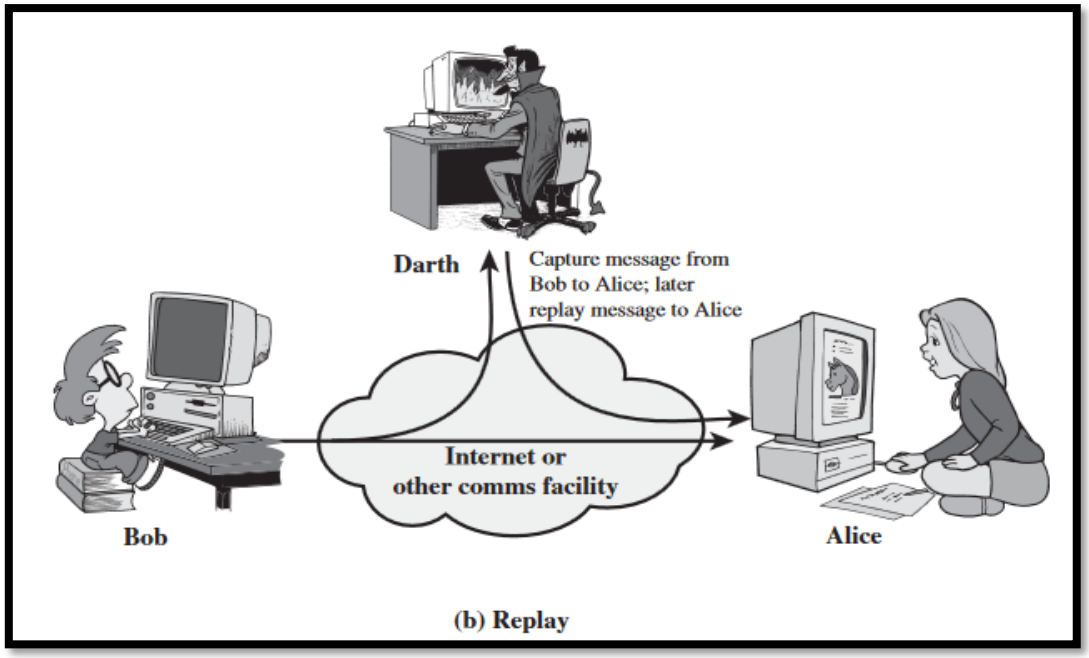



**Figure 2 :** L'attaque active (rejeu) [11]

- **L'attaque externe** L'attaquant doit utiliser la proximité physique pour pouvoir se connecter à des systèmes ou des réseaux via un accès secret ou ouvert afin de modifier, comparer et gérer l'accès aux informations.
- **L'attaque interne :** Dans ces cas d'attaques, l'attaquant peut faire partie de l'entreprise ou utiliser l'ingénierie sociale avec les personnes impliquées à la suite d'abus, de négligence ou de manque de connaissances. Dans les deux cas, ces attaques essayent d'espionner, de voler ou de détruire des informations, de les utiliser frauduleusement ou d'empêcher l'accès à d'autres utilisateurs autorisés.
- **L'attaque de distribution :** les attaques de distribution représentent toute modification malveillante du matériel ou du logiciel en usine ou lors de la distribution. Ces attaques consistent à introduire un code malveillant dans un produit comme un port dérobé pour obtenir un accès non autorisé à des informations ou une fonction système.

### 5.1.Exemples des attaques :

S'il existe des failles de sécurité dans l'hôte ou le réseau, l'attaquant utilisera certainement de nombreuses attaques pour pouvoir exploiter les exploits de l'hôte ou du réseau, nous citons par exemple [10] :

- **Craquage des mots de passe**.
- **Cheval de Troie** « Trojan Horse Un programme est caché dans un autre programme pour supprimer les soupçons, ce qui est dangereux si la victime installe le programme téléchargé, qui a portait un cheval de Troie qui va ouvrir un porte dérober au système pour certaines personnes qui utilise ce trojan.
- **IP spoofing** : L'attaquant change son adresse IP pour personnifiant l'identité d'un hôte confiant pour le permettre de bénéficier les privilèges de cet hôte afin d'accéder et manipuler avec les données critiques.
- **Les scans** : c'est la première et la plus importante étape de l'attaquant est d'obtenir suffisamment d'informations pour préparer une attaque plus efficace. Les informations pouvant être obtenues à partir de cette attaque sont le type de système d'exploitation du périphérique, les ports ouverts, etc.
- **Sniffing:** Il permet de surveiller et d'analyser le trafic réseau afin d'obtenir des informations pertinentes pour que l'attaquant peut avoir une bonne modélisation des attaques ultérieures.
- **Les attaques de déni de service (denial of service)** ont pour but de paralyser le serveur cible pour qu'il devienne inaccessible, au moins pour une durée de temps. De très nombreuses techniques existent pour épuiser les ressources d'un hôte cible, par exemple : ICMP Flooding, smurf, SYN flood, etc.
- Etc.

**6. Techniques de sécurité :**

C'est l'ensemble de procédures ou dispositifs qui sont conçu pour détecter, prévenir ou récupérer les attaques qui menacent la sécurité informatique, il existe plusieurs outils de prévention contre-attaques informatiques, Nous avons cité ci-dessous quelques mécanismes**: [11].**

- **La protection physique :** avent de parler sur la sécurité des systèmes d'information premièrement il faut assurer la sécurité des matériels informatique et leurs emplacements.
- **Chiffrement :** Les algorithmes utilisent des clés pour convertir les données afin d'obtenir une sécurité robuste. Leur sécurité dépend du niveau de sécurité des clés.

- **Signature numérique**: Un mécanisme pour assurer l'intégrité des données et également pour authentifier l'auteur du document.
- **Bourrage de trafic** : Mécanisme assurant la confidentialité des données sur le volume de trafic en cas d'interception par des attaquants.
- **Contrôle d'accès** : Vérifier l'authentification des utilisateurs et leurs autorisations d'accéder aux données et vérifier leurs privilèges.
- **Antivirus** : Logiciel censé protéger l'ordinateur contre les logiciels (ou fichiers potentiellement exécutables) néfastes. Ne protège pas contre un intrus qui emploie un logiciel légitime, ou contre un utilisateur légitime qui accède à une ressource alors qu'il n'est pas autorisé à le faire.
- **Le pare-feu** : Un élément (logiciel ou matériel) du réseau informatique contrôlant les communications qui le travers. Il a pour fonction de faire respecter la politique de sécurité du réseau, celle-ci définissant quelles sont les communications autorisées ou interdites. N'empêche pas un attaquant d'utiliser une connexion autorisée pour attaquer le système. Ne protège pas contre une attaque venant du réseau intérieur (qui ne le traverse pas).
- **Détection d'intrusion** : Identifiez une activité anormale ou suspecte sur le moniteur réseau. Ne pas détecter les accès incorrects mais autorisés par les utilisateurs légitimes. Le problème c'est comment minimiser les taux de faux positifs et de faux négatifs.
- **Journalisation** ("logs") : Enregistrement des activités de chaque acteur. Permet de constater que des attaques ont eu lieu, de les analyser et potentiellement de faire en sorte qu'elles ne se reproduisent pas.
- **Analyse des vulnérabilités** ("Security audit") : Identification des points de vulnérabilité du système. Ne détecte pas les attaques ayant déjà eu lieu, ou lorsqu'elles auront lieu **[12].**

## II. Système de détection d'intrusion :

### 1. Historique:

Les systèmes de détection ont été imposés en raison de la nécessité d'améliorer leur capacité à auditer et à surveiller la sécurité informatique. James Anderson **[13]** a été le premier à introduire le concept de systèmes de détection en 1980, mais le premier modèle a été créé par Denning Dorothy en 1987 **[14]** et plusieurs prototypes ont été produits. Des budgets importants ont investis dans la recherche dans ce domaine jusqu'à ce jour **[15].**

### 2. Définition d'un système de détection d'intrusion :

Le système de détection d'intrusion est un outil, une méthode ou une ressource utilisé pour détecter des activités non autorisées et suspectes qui peuvent être une intrusion ou non sur réseau ou dans le système d'information. Le système de détection d'intrusion est une partie importante des technologies de la sécurité informatique, mais ne constitue pas une mesure de protection indépendante. **[16]**.

Certains termes sont souvent employés quand on parle d'IDS :

− Faux positif : une alerte provenant d'un IDS mais qui ne correspond pas à une attaque réelle.

− Faux négatif : une intrusion réelle qui n'a pas été détectée par l'IDS.

3. **Classification des IDS** : Il existe plusieurs classifications des systèmes de détection des intrusions, nous avons choisi le modèle apparu dans la Figure 6 :

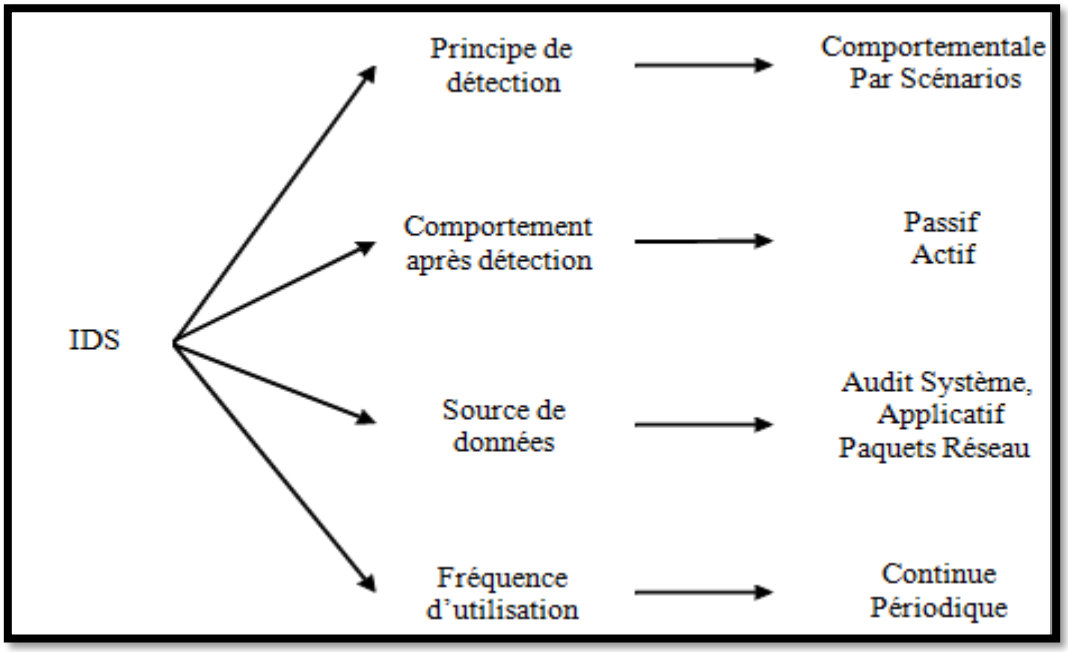



**Figure 3: Classification des IDS [17]**

### 3.1. Les méthodes de détection d'intrusion :

Le système de détection d'intrusion basé sur deux méthodes de détection : premièrement, l'approche par scénario consiste à utiliser les connaissances accumulées sur les attaques, puis à les exploiter pour prouver qu'il existe d'autres attaques, La seconde est l'approche comportementale qui consiste à créer un modèle sur la base d'une étude approfondie du comportement habituel du système ou de l'utilisateur et surveille les changements qui surviennent **[18]**.

#### 3.1.1. L'approche par scénario :

Elle repose sur un ensemble de descriptions d'attaques, également appelées signatures d'attaque, dans lesquelles ces signatures sont stockés dans la base de données, il faut que le système de détection d'intrusion contient les informations sur les vulnérabilités et cherche toute tentative de les exploiter. Dans cette approche l'IDS reçois un paquet de trafic réseau ensuite, il le compare à des modèles d'attaque pour détecter s'il est un paquet offensif ou non **[19].**

##### 3.1.1.1. Avantage [20]:

- tant que les attaques sont clairement définies à l'avance alors le taux de faux positifs est très faible.

-La détection basée sur la signature est facile à utiliser.

##### 3.1.1.2. Inconvénient [20]:

Cependant, les systèmes de détection d'intrusion par scénario comportent un certain nombre de faiblesses.

- Nécessite une connaissance spécifique du comportement d'intrusion et la collecte de données avant que l'intrusion ne soit obsolète.
- Il est difficile de détecter des attaques nouvelles ou inconnues.
- Emet des alertes quel que soit le résultat. Exemple si un ver Windows tente d'attaquer un système Linux, l'IDS envoie de nombreuses alertes d'attaque infructueuse qui pourraient être difficiles à gérer.
- Ce modèle peut ne pas toujours être aussi pratique pour les attaques internes impliquant un abus de privilèges.
- La connaissance des attaques dépend beaucoup du système d'exploitation, de la version et de l'application, et est donc liée à des environnements spécifiques.

#### 3.1.2. L'approche comportementale :

Il consiste à créer des modèles du comportement normal d'un système exploitation, les utilisateurs, les applications ou le réseau, également appelées des profils. Le profil est basé sur des métriques telles que le taux de trafic, le nombre de paquets pour chaque protocole, etc. Le profil est défini par l'administrateur ou est appris par le jeu de données lors de la phase d'apprentissage du développement d'IDS. Dans cette approche, afin de détecter des modèles d'activité anormaux, les modèles de profil prédéfinis sont comparés à ceux réellement utilisés. Les motifs détectés seront considérés comme des intrusions **[14].**

#### 3.1.2.1. Avantage [21]:

- Il a la capacité de détecter des attaques inconnues.
- Elle peut aussi détecter les attaques d'abus de privilège qui n'exploite aucune vulnérabilité.

#### 3.1.2.2. Inconvénient [21]:

- Le taux de fausses alarmes très élevé.
- Définir l'ensemble de règles pour la détection d'intrusion est difficile.
- Le comportement peut changer avec le temps. Cela nous oblige à refaire l'apprentissage le comportement normal, ce qui provoque l'indisponibilité temporaire du système de détection d'intrusion.
- Le système d'information peut subir des attaques au moment d'apprentissage.

Les systèmes de détection d'intrusion commerciaux actuels utilisent principalement une approche unique, qui est l'approche par scénario

**3.2. Comportement après la détection :** Ils existent deux types d'IDS ; actifs et passifs **[22]** :

- **IDS à réponse passive** : La réponse passive d'un IDS consiste à enregistrer les intrusions détectées dans un fichier journal qui sera analysé par le responsable de la sécurité ou générer des alarmes, envoyer un courrier électronique à un ou plusieurs utilisateurs, etc. Cela supprime les vulnérabilités de sécurité pour empêcher les attaques enregistrées de se reproduire, mais n'empêche pas directement une attaque de se produire.
- **IDS à réponse active :** La réponse active au contraire a pour but de stopper une attaque au moment de sa détection sans attendre l'intervention humaine, Pour cela on dispose de deux techniques : la reconfiguration du firewall et l'interruption d'une connexion TCP,

- La reconfiguration du firewall permet de bloquer le trafic malveillant au niveau du firewall, en fermant le port utilisé ou en interdisant l'adresse de l'attaquant. Cette fonctionnalité dépend du modèle de firewall utilisé, tous les modèles ne permettant pas la reconfiguration par un IDS. De plus, cette reconfiguration ne peut se faire qu'en fonction des capacités du firewall.
- L'IDS peut également interrompre une session établie entre un attaquant et sa machine cible, de façon à empêcher le transfert de données ou la modification du système attaqué.

**3.3. La source des données** : Les IDS peuvent être classés en fonction de la provenance de leurs données d'audit, selon qu'elles viennent du système, des applications, des paquets du réseau ou encore d'un autre IDS **[17].**

**3.4. La fréquence d'utilisation** : La fréquence d'analyse des données d'audit est aussi un élément distinctif des systèmes de détection d'intrusions. Certains IDS peuvent surveiller en permanence le système d'information tandis que d'autres se limitent à une analyse périodique **[17].**

**4. Les types des IDS :**

Il existe plusieurs types d'IDS disponibles aujourd'hui, car ils sont jouent un rôle important dans la capacité de survie du système d'information et préservent sa sécurité des attaques. Un IDS peut être classé comme :

**4.1.L'IDS basé hôte (host- based IDS):**

HIDS est un système de détection d'intrusion spécifique à un ordinateur unique qui surveille la sécurité de ce système ou de cet ordinateur contre les attaques internes et externes. Les attaques internes font référence au cas où il détecte quel programme a accès à quelle ressource et qu'il y a une faille de sécurité. Dans la deuxième partie, il s'agit d'attaques externes, HIDS analyse les paquets en provenance et à destination de ce système (ordinateur) sur ses interfaces. HIDS répond en enregistrant l'activité et en informant l'autorité désignée **[23]**.

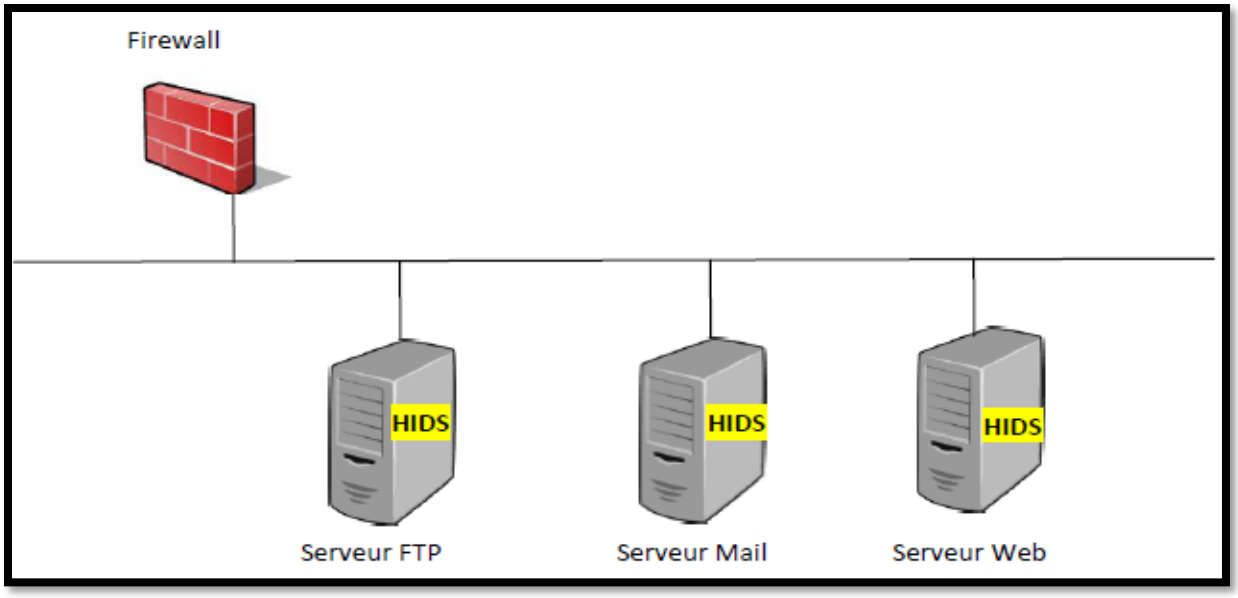



**Figure 4 :** Exemple d'un HIDS (L'IDS –Niveau Système) **[24]**

Les avantages et les inconvénients **[23]** de l'IDS basé hôte sont :

#### 4.1.1. Les avantages d'un IDS basé hôte

- HIDS peut protéger du réseau local.
- HIDS est polyvalent.
- Nécessite moins de formation que le NIDS.
- HIDS fournit une analyse du registre de la machine locale.

#### 4.1.2. Les inconvénients d'un IDS basé hôte

- Système passif qui doit attendre qu'un événement soit une indication d'une attaque et ne peut pas l'empêcher de manière proactive.
- La collecte de données s'effectue par hôte.
- L'écriture dans le journal ou l'activité de génération de rapports générera une charge supplémentaire pour le réseau.
- Les pirates informatiques intelligents peuvent attaquer et désactiver HIDS, tandis qu'attaquer HIDS consomme du temps de traitement, du stockage, de la mémoire et d'autres ressources système.

### 4.2.L'IDS basé réseau (Network- based IDS):

Le système de détection d'intrusion basée réseau (NIDS) surveille le trafic réseau et analyse les paquets en transit pour détecter les attaques. Lors de l'identification d'une attaque ou lorsqu'un comportement anormal est détecté, une alerte peut être envoyée à l'administrateur **[23]**.

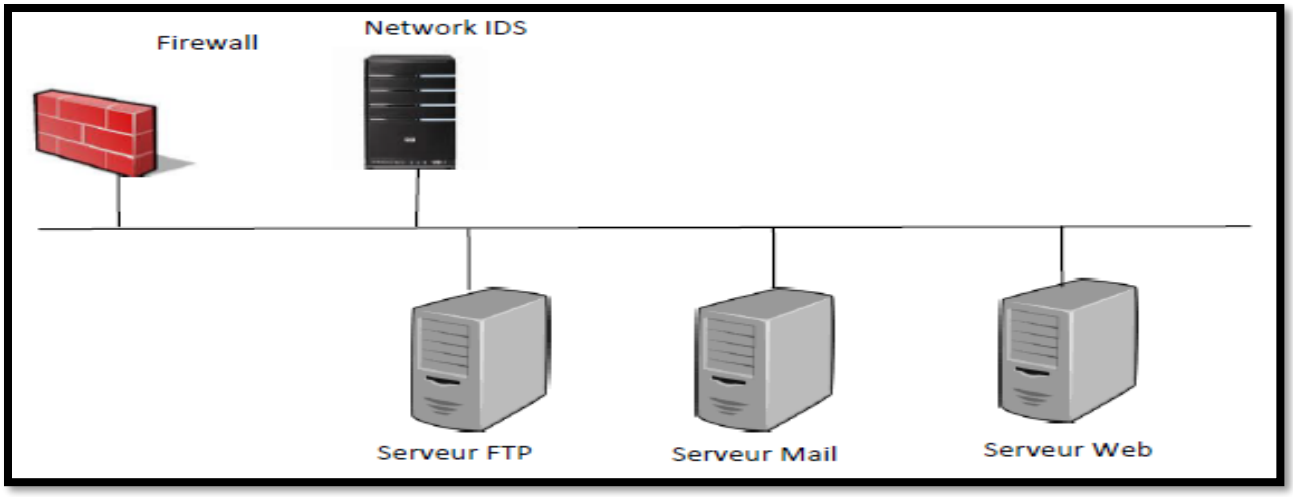



**Figure 5 :** Exemple d'un IDS dans un réseau (NIDS) **[24]**

Les avantages et les inconvénients **[23]** de ce type d'IDS sont :

#### 4.2.1. Les avantages d'un IDS basé réseau

- Adaptable à l'environnement multiplateforme.
- Le NIDS est géré de manière centralisée.
- L'IDS basé sur le réseau offre une grande sécurité contre les attaques, car il est invisible aux attaquants.
- L'IDS basé réseau est capable de contrôler un grand nombre d'hôte.

#### 4.2.2. Les inconvénients d'un IDS basé réseau

- Nécessite plus de formation.
- Utilise la bande passante LAN.
- Le taux d'échec est plus élevé.

### 4.3.L'IDS hybride :

Généralement utilisés dans un environnement décentralisé, ils permettent de réunir les informations de diverses sondes placées sur le réseau. Leur appellation « hybride » provient du fait qu'ils sont capables de réunir aussi bien des informations provenant d'un système HIDS qu'un NIDS.

## 5. L'Architecture de base des IDS :

Plusieurs architectures ont été proposées pour décrire les différents éléments intervenants dans un système de détection d'intrusion. L'architecture la plus simple est composée de trois modules : la source de données, l'analyseur des données et le module des réponses **[14]** voir figure 6.

- **Les capteurs :** sont responsables de la collecte des données. La source de données pour un capteur peut être n'importe quelle partie d'un système pouvant contenir des preuves

d'une intrusion. Les données peuvent être des paquets réseau, les fichiers journaux et les traces d'appels système. Les capteurs collectent et transmettent ces informations à l'analyseur **[25].**

- **L'analyseur des données :** Les analyseurs reçoivent les informations d'un ou plusieurs capteurs ou d'autres analyseurs. L'analyseur est responsable de déterminer si une intrusion s'est produite. La sortie de ce composant indique qu'une intrusion s'est produite. Le résultat peut inclure des éléments de preuve permettant de conclure à une intrusion. L'analyseur peut fournir des indications sur les mesures à prendre à la suite de l'intrusion **[25].**
- **Le module de réponses :** C'est le module qui assure les réponses face aux activités malveillantes détectées. Les réponses peuvent être actives ou passives, en fonction des contre-mesures entamées pour contrer les intrusions. Ça peut être un simple message d'alerte, ou une sauvegarde dans un fichier log ou bien interrompre une connexion **[25].**

Ces trois modules sont communs à la majorité des architectures proposées dans la littérature.

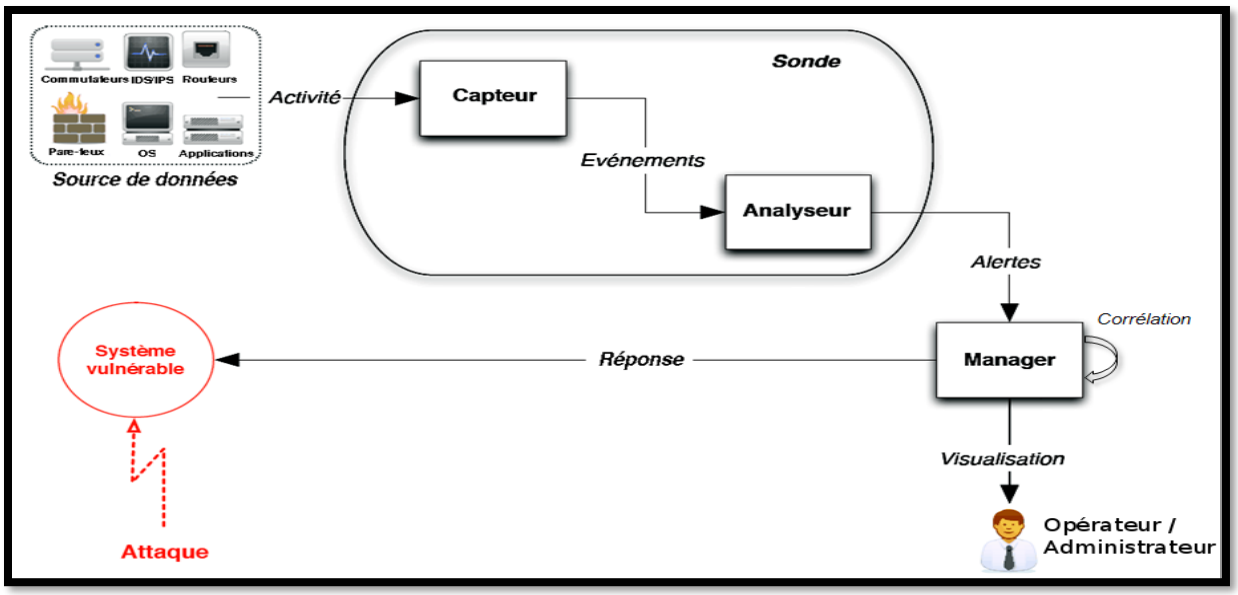



**Figure 6:** Architecture de base d'un IDS **[14]**

### 5.1.Les Architectures d'implémentation des IDS :

L'architecteur d'implémentation c'est la stratégie qui décrit les éléments d'un IDS. On distingue trois approches d'implémentation :

#### 5.1.1. L'approche monolithique (centralisée) :

Les premiers systèmes de détection d'intrusions ont employé une architecture monolithique, elle se composent de plusieurs moniteurs qui surveillent le comportement de leur hôte respectif ou le trafic réseau passant par. Ces moniteurs partagent leurs données avec une unité d'analyse centrale. Cependant, cette approche ne peut pas supporter un grand réseau à cause de la quantité énorme des données des différents hôtes qui doivent être analysée par le serveur central, ce qui engendre une dégradation sévère des performances de réseau **[26].**

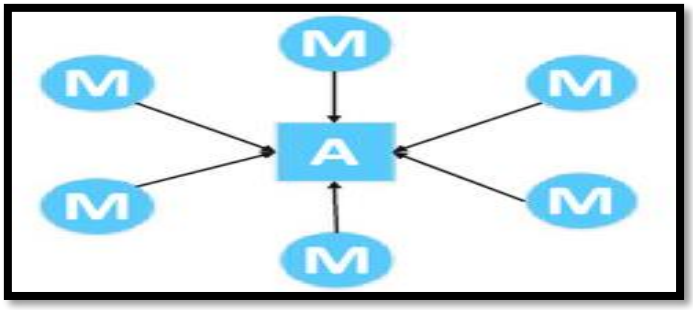



**Figure 7:** Architecture d'IDS centraliser **[26]**

Parmi les IDS basés sur l'approche centralisée NADIR **[27].**

#### 5.1.2. L'approche hiérarchique

Cette approche a été proposée pour surmonter les problèmes de l'approche monolithique. Elle est caractérisée par l'existence des secteurs de contrôle hiérarchiques. Chaque IDS contrôle un secteur avec l'élimination du transfert des données d'audit rassemblées par les hôtes locaux à un point central. Chaque IDS à n'importe quel niveau de contrôle exécute une analyse locale et envoie ses résultats d'analyse au niveau suivant dans la hiérarchie [26].

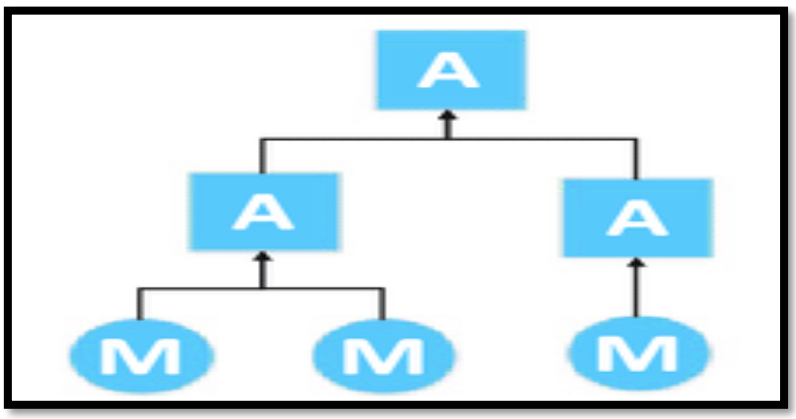



**Figure 8:** Architecture d'IDS hiérarchique **[26]**

Un exemple de système de détection d'intrusions hiérarchique : EMERALD **[28].**

### 5.1.3. L'approche coopérative (distribuée)

Cette approche a été suggérée pour résoudre les problèmes de l'approche précédente. Elle essaye de distribuer les responsabilités d'un serveur central à un nombre de systèmes de détection d'intrusions coopératifs. La différence de cette approche avec l'approche hiérarchique est qu'il n'y a aucune hiérarchie entre les IDS distribués ce qui signifie que l'échec de n'importe quel IDS n'empêche pas la détection d'attaques coordonnées **[26].**

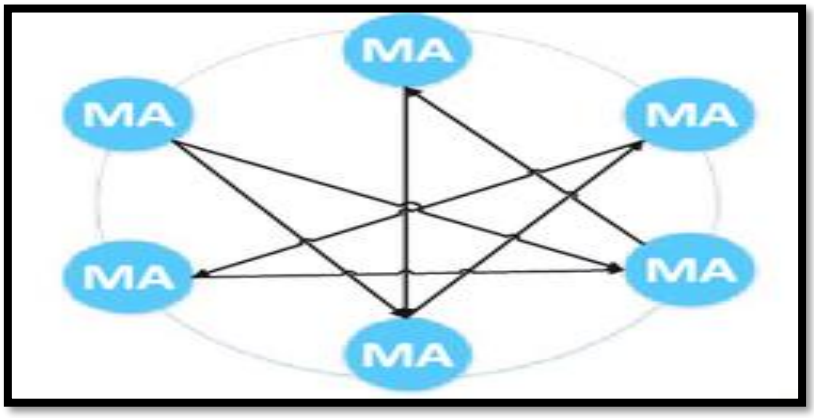



**Figure 9:** Architecture d'IDS distribuer **[26]**

Parmi les systèmes de détection d'intrusions coopératifs, nous pouvons citer par exemple le système AAFID [29].

## 6. Conclusion :

Dans ce premier chapitre nous présentons deux parties, la première présente les principales notions de base de la sécurité informatique, en commençant par les définitions des différentes notions de la sécurité informatique, puis les attaques informatiques et leurs classifications avec exemples. La deuxième partie présente les systèmes de détection d'intrusions, leur historique, définition, et Ses critères de classification, ...etc.

# *Chapitre 2 :*

## Les systèmes multi-agent

## 1. Introduction :

IDS est une solution de gestion de la sécurité intégrée à de nombreuses approches d'intelligence artificielle **[30]**, mais la plupart d'entre elles ont été complexes, aussi la distribution des hôtes rendre la détection d'intrusion difficiles, Ce qu'il faut, une solution de sécurité flexible et adaptable offrant une plus grande autonomie. Il est donc nécessaire de revoir la manière dont la détection d'intrusion standard est conçue et implémenter pour identifier et atténuer sa vulnérabilité. Dans ce contexte, les systèmes multi-agents offrent un équilibre entre les exigences de sécurité, la flexibilité du système et l'adaptabilité. En fait, la technologie des systèmes multi-agents (SMA) est l'un des domaines de l'intelligence artificielle distribuée (DAI) qui consiste en un ensemble de facteurs individuels appelés environnements distribués. Chaque agent coopère et communique avec d'autres agents.

Dans ce chapitre, on parle sur l'approche de SMA intégrer avec la détection d'intrusion premièrement on définir l'agent avec ces caractéristiques et leurs types, et on a également étudié le système multi-agents. Enfin, on cite quelques systèmes de détection d'intrusion basé sur le système multi-agent.

## 2. L'Intelligence Artificielle et L'Intelligence Artificielle Distribuer :

- **L'Intelligence Artificielle (IA) :** L'intelligence artificielle a été proposée pour la première fois par John McCarthy en 1956 **[30]**. L'IA est le domaine d'étude décrivant la capacité de l'apprentissage automatique, au même titre que l'être humain, et la capacité de réagir à certains comportements, Ces processus ont été appliqué dans plusieurs domaines tels que les réseaux de neurones, algorithmes génétiques, systèmes experts, système multi-agent **[31]** ….

- **L'Intelligence Artificielle Distribuer :** c'est une nouvelle voie de l'IA, il est introduit le concept du système Multi-Agents qui se caractérise en mention de collaboration et fonctionne par la coopération, la coordination et la communication **[32]**.

## 3. Quelques techniques de l'IA appliquées au IDS :

- **Réseaux de neurones artificiels (ANN) :** est inspiré de système nerveux de l'homme, qui est connecté par l'intermédiaire de neurones. Les réseaux de neurones ont la capacité de comprendre et d'apprendre par la formation et peuvent être utilisés pour identifier des tendances complexes **[33]**.

- **Algorithme génétique :** sont largement utilisés dans de nombreux domaines de l'informatique pour résoudre un problème complexe. Il fournit des solutions robustes, adaptatives et optimales à de nombreux problèmes informatiques. Les algorithmes génétiques en informatique sont inspirés de biologiques des processus tels que la sélection naturelle, l'évolution, la théorie de la mutation et le patrimoine génétique **[33]**.

- **L'approche statistique :**

Elle consiste à mesurer le comportement de l'utilisateur ou du système par un nombre de variables échantillonnées dans le temps. Ces variables comprennent **[34]** :

- Le temps de connexion et de déconnexion de chaque session.
- L'utilisation de la mémoire.
- L'occupation du processeur.
- L'accès aux fichiers.

- **Data mining :** est l'extraction automatique de données qui n'avaient pas encore été réaliser à partir de sources de données volumineuses dans le but de prendre en charge des actions. Le développement rapide récent de l'exploration de données a mis à disposition une grande variété d'algorithmes, issus des domaines de la statistique, de la reconnaissance de formes, de l'apprentissage par machine et des bases de données. Plus précisément, les approches de data mining ont été utilisées pour la détection d'anomalies **[35]**.
- **Machine Learning :** Cette approche aide à l'extraction automatique des caractéristiques des activités normales qui sont critique pour la détection d'anomalies. A partir des données d'audit, la machine Learning identifie des règles pour définir le comportement normal et détermine si un nouvel événement observé est anormal ou non **[36]**.
- **L'immunologie**

Le système immunitaire artificiel (AIS) est un ensemble d'algorithmes inspirés des principes et des fonctions du système immunitaire biologique. Ce dernier exploite les caractéristiques du système immunitaire naturel, en ce qui concerne l'apprentissage et la mémorisation afin de résoudre des problèmes complexes dans le domaine de l'intelligence artificielle **[37]**.

- **Système Multi-agent :** Les agents sont capables de réaliser des traitements simultanés, sont capables de s'auto-adapter à l'évolution de l'environnement et ont également la propriété de la coordination distribuée.

- Les systèmes multi-agents peuvent être considérés comme un ensemble d'entités artificielles autonomes capables d'exécuter diverses tâches par le biais de l'interaction, de la coordination, de la communication, de l'intelligence collective et de l'émergence de modèles de comportement **[37]**

Plusieurs approches ont été utilisées dans le système de détection d'intrusion, y compris le système multi-agents qu'on s'intéressés à étudier, car il présente des caractéristiques permettant une modélisation améliorée de la sécurité dans les systèmes de détection d'intrusion.

**4. Agents et systèmes multi agent :**

Les systèmes multi-agents, en tant que sous-domaines de DAI, sont considérés comme des systèmes informatiques dans lesquels plusieurs agents autonomes et intelligents interagissent et travaillent en collaboration pour effectuer un ensemble de tâches et atteindre un ensemble d'objectifs.

**4.1.Définition d'Agent :**

**Définition 1 :** « Un agent est une entité autonome, capable d'agir sur elle-même et sur son environnement, et dont ces actions sont les conséquences de ces observations, ces connaissances, son interaction et communication avec d'autres agents » **[38]**.

**Définition 2 :** « Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu » **[39]**.

**4.2.Les caractéristiques des agents** : A partir des définitions précédentes nous pouvons décrire plusieurs propriétés qui caractérisent le comportement des agents **[40] [41]**

- **Intelligence:** le terme « intelligence » signifie que l'agent est en mesure d'afficher un niveau de priorité de renseignement différent, allant d'actions prédéfinies (planification) à l'auto-apprentissage (définition de nouvelles actions).
- **Autonomie:** est la capacité d'un agent de fonctionner sans intervention directe d'êtres humains ou d'autres agents et d'avoir un contrôle quelconque sur son état interne et son environnement externe ;
- **Situer:** l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement. Exemples: systèmes de contrôle de processus, systèmes embarqués.

- **Communication:** Pour que les agents puissent collaborer et coordonner leurs actions à effectuer, ils doivent échanger des messages entre eux.
- **Adaptabilité:** Les agent peuvent s'adapter au changement de leur environnement et ont la capacité d'acquérir un comportement intelligent par l'apprentissage.
- **Sociabilité :** est la capacité d'un agent à s'intégrer dans un vaste environnement peuplé d'une société d'agents avec laquelle l'agent doit échanger des messages pour réaliser des actions utiles. Cette propriété est satisfaite même lorsque les systèmes doivent partager leurs connaissances et leurs attitudes mentales (croyances, objectifs, désirs, etc.).
- **Proactivité:** est la capacité d'un agent d'anticiper les situations et de changer de ligne de conduite. C'est une propriété pertinente qui apparaît dans la gestion du réseau et du système afin d'éviter des effets désastreux sur les performances globales. En effet, les agents proactifs sont capables de manifester des comportements axés sur les objectifs en prenant certaines initiatives.
- **Réactivité:** est la capacité d'un agent de modifier son comportement au fil du temps pour atteindre ses objectifs en matière de résolution de problèmes.

### 4.3.Les types des agents :

Les actions des agents sont prises en fonction de perceptions retenues par les situations externes. Dans intelligence artificielle distribuer (DAI) on distingue trois types d'agents en fonction de leur niveau d'intelligence réactif, cognitif:

- **Agents réactifs:** Un agent réactif réagit rapidement pour résoudre un problème simple qui ne nécessite pas de raisonnement complexe. Ainsi, l'intelligence du système émerge des interactions entre un grand nombre de ce type d'agents **[42]**.

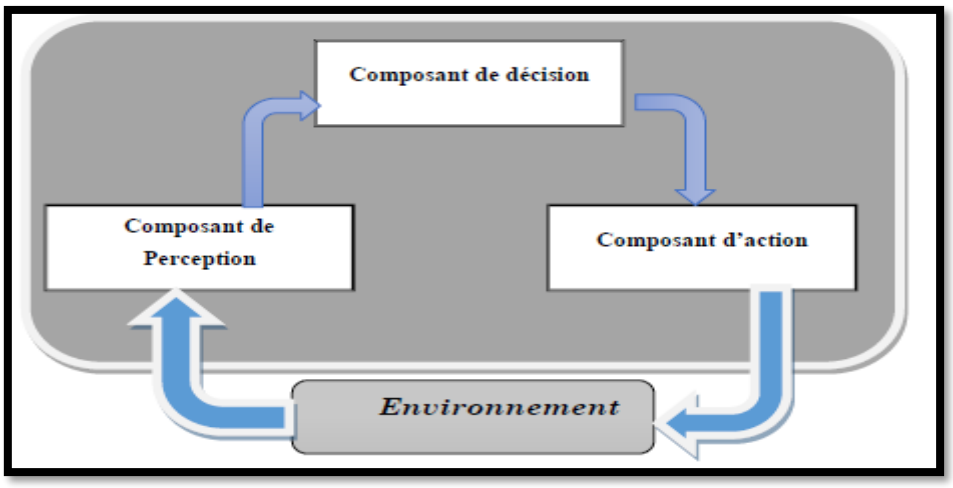



**Figure 10 :** Structure d'un agent réactif [**43]**

- **Agents cognitifs:** un agent cognitif est capable de trouver une solution à un problème complexe tout en communiquant avec d'autres agents et en interagissant avec sa base de connaissances. Ses principales caractéristiques comprennent une grande capacité de raisonnement, le traitement de données, la perception, l'apprentissage, le contrôle, la communication et le domaine de la réactivité de l'expertise **[42]**.

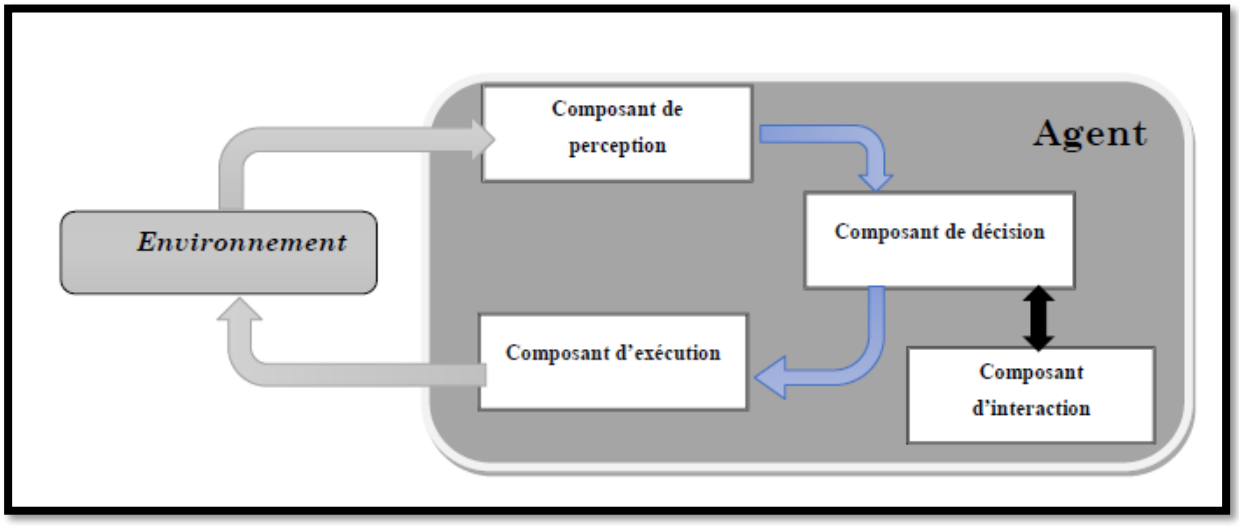



**Figure 11 :** Structure d'un agent cognitif **[43]**

- **Agents hybrides:** Est un mélange d'agent réactif et cognitif, possède un réflexe (évolution réactive) pour résoudre des problèmes répétés et réfléchit (une attitude cognitive) à des situations système complexes **[42]**.

**5. Systèmes Multi Agents :**

Jacques Ferber **[41]** propose une définition plus précise d'un SMA :

**Définition:** « On appelle système multi-agent (ou SMA), un système composé des éléments suivants :

- Un environnement **E**, c'est-à-dire un espace disposant généralement d'une métrique.
- Un ensemble d'objets **O**. Ces objets sont situ´es, c'est-`a-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans **E**. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
- Un ensemble **A** d'agents, qui sont des objets particuliers (**A⊆O**), lesquels représentent les entités actives du système.
- Un ensemble de relations **R** qui unissent des objets (et donc des agents) entre eux.
- Un ensemble d'opérations **Op** permettant aux agents de **A** de percevoir, produire, consommer, transformer et manipuler des objets de **O**.

- Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, quel 'on appellera les lois de l'univers. »

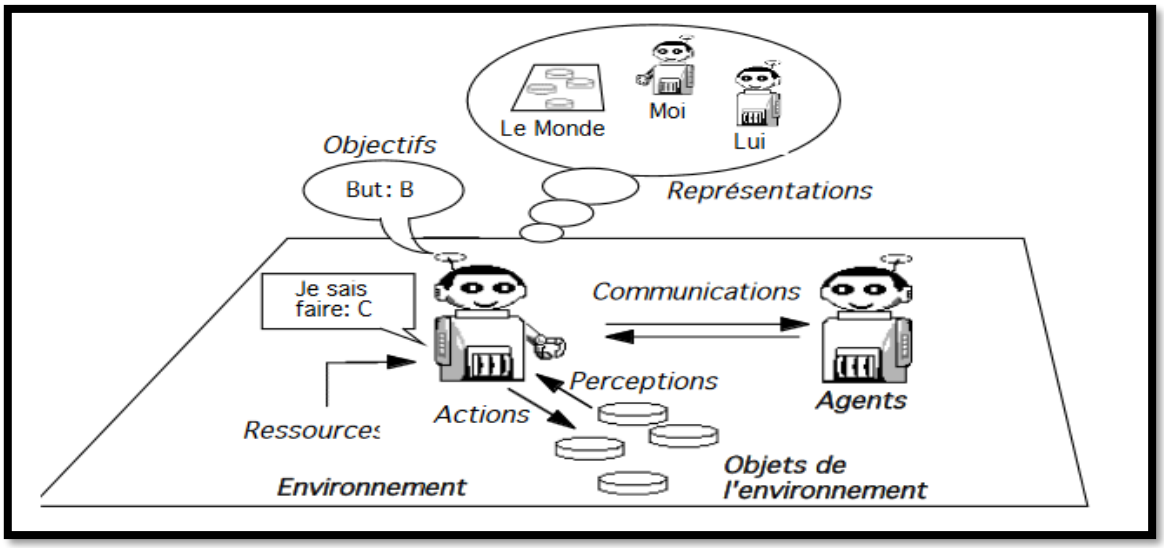



**Figure 12 :** Représentation du Système Multi Agents **[41]**

## 6. Propriétés des systèmes multi-agent:

- **Coopération :** Ce sont des tâches effectuées par un groupe d'agents pour accroître l'efficacité du système **[44]**.
- **Coordination** permet d'assurer la cohérence des actions exécutées par des agents. Parmi les techniques de coordination : l'organisation, la planification et la synchronisation **[45]**.
  - **L'organisation** forme des sous-ensembles d'agents travaillant pour un but précis.
  - **La planification** consiste à la création d'un plan multi-agents, qui réalise l'ordonnancement, l'interaction et l'exécution des tâches entre les agents. Ce plan se déroule en trois phases.
    - **La synchronisation** englobe les mécanismes de déroulement temporel des actions.
    - **Délégation** Cette propriété est très importante car elle permet à un agent d'exécuter les tâches d'un autre agent dont les ressources sont limitées.
    - **Communication** Les agents communiquent entre eux par échange de messages afin de bien mener des actions communes liées aux buts à atteindre.

- **Négociation** est une partie importante du travail de coordination, car la négociation est une technique clé de coordination utilisée pour résoudre plusieurs problèmes de DAI. La négociation est considérée comme un processus de communication d'un groupe d'agents en vue de parvenir à un accord mutuellement accepté sur un problème et pour négocier efficacement, les agents doivent raisonner sur les croyances, les désirs et les intentions d'un autre agent.

7. **Classification des systèmes multi-agents :**

Les systèmes multi-agents sont classés en quatre grandes familles selon certains critères : le nombre d'agents, la nature ou bien la complexité des agents **[46]** :

- **SMA Ouvert** : dans ce type de système en peut ajouter ou supprimé des agents.
- **SMA Fermé** : ce type de système est statique, le nombre des agents ne change pas.
- **SMA Homogène** : un seul modèle est utilisé pour générer tous les agents intervenant dans le système.
- **SMA Hétérogène** : les modèles d'agents sont différents.

**8. Intérêts et avantages des SMA :**

Les systèmes Multi-Agents présentent des intérêts irréfutables pour la modélisation et la gestion des architectures distribuées. L'avantage des SMA réside dans sa modularité, sa stabilité et sa robustesse **[32]**:

- La modularité permet de partitionner le problème en plusieurs sous-ensembles d'agents ce qui permet de réduire la complexité.
- La tolérance aux erreurs suite au contrôle réparti entre les agents permet d'influencer le comportement global du système.
- La coordination entre les agents rend les résultats globaux plus importants que les résultats locaux.
- La distribution des traitements s'adapte avec les problèmes liés aux systèmes.
- La diversité des données traitées rend les champs d'applications très larges et facilite les extensions futures.
- La robustesse et la stabilité des performances permettent la prise de décision collaborative.

9. **Detection d'intrusion à l'aide d'agents :**

IDS peut utiliser des agents autonomes, des agents intelligents, des agents mobiles ou une combinaison de ces agents. On utilise les principes de différents domaines, tels que

l'intelligence artificielle, les réseaux de neurones, la logique floue, les algorithmes génétiques, etc. pour rendre les agents intelligents.

**9.1.Un agent intelligent :**

Un agent capable de prendre des décisions en fonction de son expérience. ... Par conséquent, un agent intelligent autonome et artificiel est capable d'effectue des actions basées sur les informations qu'il perçoit, selon ses propres expériences et ses propres décisions. [47].

**9.2.Agents autonomes:**

Une entité logicielle qui fonctionne de manière continue et autonome dans un environnement particulier ... capable de mener des activités de manière flexible, intelligente et sensible aux changements de l'environnement. Idéalement, un agent qui fonctionne en permanence pourrait apprendre de son expérience. De plus, nous nous attendons à ce qu'un agent qui habite un environnement avec d'autres agents et processus soit capable de communiquer et de coopérer avec eux, et peut-être se déplacer d'un endroit à l'autre [48].

**9.3. Agent Mobile :**

Un agent mobile est capable de se déplacer au cours de son exécution dans le réseau, d'un site à un autre pour accéder à des données (ou des ressources) ou à la demande d'un client (autre agent ou humain). Il se déplace avec ses données propres et son code, ainsi qu'avec son état d'exécution. C'est un paradigme de plus en plus utilisé dans les systèmes distribués **[49]**.

9.3.1. **Avantages à l'usage d'agents mobiles dans la détection d'intrusion :** Plusieurs avantages liés à l'utilisation d'agents mobiles sont décrits dans la littérature **[56] [57] [58]**. Sont présentés ci-dessous:

- **Retard causé par les réseaux:** Lorsque des IDS hiérarchiques sont utilisés dans un réseau, la réponse est plus lente en cas d'attaque. En effet, le contrôleur central (machine) doit envoyer les informations relatives à l'attaque à chaque hôte participant et la réponse doit être donnée par chaque hôte particulier du réseau. Cela peut ne pas toujours entraîner une réponse immédiate, car les informations peuvent mettre trop de temps à atteindre l'hôte de destination. Ainsi, les IDS hiérarchiques traditionnels peuvent ne pas réussir à détecter rapidement les attaques. En revanche, si des agents mobiles sont utilisés, ils peuvent répondre plus rapidement car ils sont directement envoyés du contrôleur central à l'hôte cible.
- **Minimiser le trafic réseau:** les systèmes IDS traditionnels utilisent différents mécanismes de collecte de données pour collecter des données à la fois au niveau de l'hôte et du réseau. Ces données sont ensuite utilisées pour suivre toute intrusion. En règle générale, la quantité de données collectées est très importante et, pour qu'une

intrusion soit détectée, les données provenant de différents hôtes doivent être collectées et traitées par le contrôleur central. Cela augmente le trafic sur le réseau, créant ainsi une surcharge sur le réseau. En utilisant des agents mobiles, la charge sur le réseau peut être réduite car ces agents mobiles utilisent des mécanismes de recherche efficaces, réduisant ainsi la nécessité d'un trafic de données entre plusieurs hôtes.

- **Persistance:** étant donné que les nœuds mobiles fonctionnent de manière autonome et asynchrone, ils ne sont pas sujets aux pannes, même en cas d'échec de la machine qui les héberge. Sur les machines centralisées, en cas de défaillance du contrôleur central, l'IDS dans son ensemble est considéré comme étant en panne car il n'ya pas de communication entre les autres hôtes.
- **Indépendance de la structure et de la plate-forme:** Les agents mobiles peuvent être utilisés dans les IDS avec une structure flexible. Par exemple, un agent peut être désigné pour collecter les données sur le réseau, un autre agent peut être utilisé pour détecter et signaler des anomalies et le reste peut être utilisé pour prendre les mesures appropriées. En raison de cette structure, les agents mobiles trouvent une application formidable dans les IDS. En outre, des agents mobiles de différents fournisseurs peuvent être utilisés pour créer des IDS. Il est également possible d'écrire votre propre code mobile pour le rendre applicable à l'environnement existant.
- **Nature dynamique:** la nature dynamique des agents mobiles permet de les déplacer sur le réseau. Cela permet également de reconfigurer le système pendant l'exécution. Les agents mobiles peuvent être clonés, distribués ou mis en veille lorsque la configuration du réseau doit être modifiée. En outre, ils peuvent détecter leur environnement d'exécution et s'adapter dynamiquement à la situation.
- **Hétérogénéité d'environnement:** les agents mobiles peuvent être interopérables sur plusieurs plates-formes en raison de l'interpréteur virtuel installé sur la machine hôte. Les agents mobiles sont généralement indépendants de l'ordinateur et de la couche de transport et ne dépendent que de l'environnement d'exécution. Cette fonctionnalité permet aux agents mobiles d'être utilisés sur plusieurs plates-formes différentes sans problèmes de compatibilité.
- **Nature robuste :** même en cas de défaillance d'un des agents, les autres agents de l'IDS peuvent prendre en charge les tâches de l'agent défaillant et poursuivre la détection. Ce comportement robuste des agents mobiles les rend plus applicables dans les grands

environnements où plusieurs agents et leur interaction sont nécessaires pour une surveillance adéquate du réseau.

- **Scalabilité :** En utilisant les IDS d'agent mobile distribués, il est plus facile de gérer de grands réseaux. Les agents ont la capacité de se cloner et de se distribuer eux-mêmes sur les nouvelles machines lorsqu'elles sont ajoutées au réseau.

**10. Travaux connexes :**

**10.1. Autonomous Agents for Intrusion Detection :**

L'effort AAFID (agents autonomes pour la détection d'intrusion) chez Purdue **[50]** est à bien des égards un IDS classique avec des agents utilisés principalement pour structurer le composant de collection de détection d'intrusion en un ensemble de composants logiciels légers facilement reconfigurables. L'AAFID utilise une hiérarchie d'agents. Des moniteurs se trouvent à la base de la hiérarchie. Ils fournissent un ordre et un contrôle globaux et analysent les informations émanant des nœuds de niveau inférieur. Aux feuilles se trouvent des agents qui collectent des informations sur les événements. Les agents résident sur des plates-formes d'agents à usage spécifique, appelées émetteurs-récepteurs. Les émetteurs-récepteurs commandent et contrôlent les agents locaux et analysent ou réduisent le traitement des informations reçues des agents. Les émetteurs-récepteurs transmettent les informations traitées sur des moniteurs. Les agents semblent être statiques une fois déployés sur un émetteur-récepteur, mais peuvent également être remplacés par reconfiguration.

**10.2. Hummingbird :**

L'Université de l'Idaho a développé le projet Hummingbird **[51]**. C'est l'un des prototypes de détection d'intrusion distribuée les plus ambitieux disponibles. Le système Hummingbird est un système distribué permettant de gérer les données relatives à une mauvaise utilisation. Bien que le système utilise une technologie d'agent, les agents ne sont ni autonomes ni mobiles. Seule la collecte de données est distribuée et le contrôle reste centralisé. L'accent est mis sur le partage de données relatives à la sécurité entre des sites ayant différents domaines de sécurité. Les outils, algorithmes, techniques de réduction des données et de visualisation offrent des perspectives considérables pour une utilisation dans un système d'agent mobile. Hummingbird n'implémente pas de nouvelles fonctionnalités de sécurité pour se protéger. Au lieu de cela, il s'appuie sur le système Kerberos **[52]**.

**10.3. Java Agents for Meta-Learning :**

Le projet Java Agents for Meta-Learning (JAM) **[53]** de l'Université de Columbia, dans l'État de New York, applique le méta-apprentissage à l'exploration de données distribuée à l'aide d'agents intelligents. Les agents intelligents utilisent des techniques d'intelligence artificielle

pour modéliser les connaissances et le raisonnement, ainsi que le comportement, dans des sociétés ou des domaines multi-agents. La conception comporte deux composants clés: des agents de détection de fraude locaux qui apprennent à détecter une fraude et à fournir des services de détection d'intrusion au sein d'un système d'information d'entreprise unique, et un système de méta-apprentissage sécurisé et intégré qui combine les connaissances collectives acquises par des agents locaux individuels. L'exploration de données, à l'instar des réseaux de neurones et d'autres applications d'apprentissage à point unique, ne permet pas le partage de connaissances entre agents. La méthode du méta-apprentissage tente de surmonter cette limitation en intégrant un certain nombre de classificateurs appris séparément, intégrés sous forme d'agents distants.

### 10.4. Intelligent Agents for Intrusion Detection :

Ce projet de l'Université d'État de l'Iowa **[54]** implique un système IDS basé sur la technologie des agents intelligents, d'une manière quelque peu similaire à celle de JAM. La mobilité des agents permet à différents types d'agents intelligents utilisant des algorithmes de classificateur de se déplacer entre des points de collecte, appelés « nettoyeurs de données », et de détecter des activités suspectes. Les algorithmes d'agent sont des méthodes d'identification de séquence standard et des mécanismes de détection d'identification de vecteur de caractéristiques. L'architecture est hiérarchique, avec un entrepôt de données à la racine, des nettoyeurs de données aux feuilles et des agents de classification entre les deux. Un agent classificateur est spécialisé dans une catégorie d'intrusion spécifique et est capable de collaborer avec des agents d'une autre catégorie pour déterminer le niveau de gravité d'une activité considérée comme suspecte. Le déplacement de l'analyse informatique (l'agent de classificateur, par exemple) vers chaque point de collecte évite le déplacement coûteux d'informations vers une unité d'agrégation. Les résultats fournissent une bonne base pour les travaux ultérieurs, étant donné que l'approche établit la capacité de définir des agents de détection d'intrusion qui ciblent des systèmes et des sous-systèmes individuels.

### 10.5. Advanced Telecommunications/Information Distribution Program :

Les travaux en cours dans le cadre du programme de recherche sur les télécommunications avancées / distribution de l'information (ATIRP) **[50]** traitent des vulnérabilités informatiques de détection utilisant des agents de gestion, et non de la détection d'intrusion. Cependant, les modules de détection d'intrusion pourraient facilement être remplacés par des modules d'évaluation de la vulnérabilité afin de créer un IDS rudimentaire. Un répartiteur central lance des agents sur un ou plusieurs nœuds cibles pour tester les vulnérabilités connues et signaler les résultats. Les agents sont composés de manière dynamique à l'aide d'un algorithme génétique,

qui tente en permanence de maximiser la probabilité de découvrir des vulnérabilités existantes. Le pool de gènes à partir duquel les agents évoluent est constitué de fragments de code correspondant à une technique de détection et conçus pour être composés avec d'autres fragments. L'architecture dispose de capacités de sécurité importantes et repose sur des signatures cryptographiques et des certificats de clé publique. Des capacités identiques ou similaires seraient nécessaires dans un IDS. Le système devrait être étendu pour gérer un système IDS, car la communication entre agents est plus essentielle pour la détection des intrusions que pour le balayage des vulnérabilités.

### 10.6. Intrusion Detection Agent System (IDA) :

Au Japon, l'Agence de promotion des technologies de l'information (IPA) met au point un système IDS appelé système de détection d'intrusion (IDA) **[55]**. L'IDA est un IDS basé sur plusieurs hôtes. Au lieu d'analyser l'ensemble des activités des utilisateurs, IDA surveille des événements spécifiques pouvant être liés à des intrusions, appelés marques laissées par une intrusion suspectée (MLSI). Si un MLSI est trouvé, l'IDA recueille des informations relatives au MLSI, les analyse et décide s'il y a eu ou non une intrusion.

Le système IDA s'appuie sur des agents mobiles pour localiser les intrus parmi les différents hôtes impliqués dans une intrusion et pour collecter des informations. L'architecture est hiérarchique, avec un gestionnaire central à la racine et une variété d'agents aux feuilles. Un capteur est un agent qui réside sur un nœud à la recherche de MLSI. Lors de la découverte de telles informations, le capteur informe le gestionnaire qui envoie un agent de traçage à l'hôte. L'agent de traçage initie un agent de collecte d'informations pour recueillir des informations connexes sur l'hôte, avant déplacer sur tout autre site identifié comme un point d'origine présumé. Le gestionnaire collecte et intègre les résultats de l'agent de collecte d'informations à leur retour. La duplication possible causée par plusieurs capteurs détectant la même intrusion est résolue par un tableau de messages sur chaque hôte surveillé. Les développeurs indiquent que le système multi-agents obtenu est un moyen efficace de détecter les intrusions.

**Conclusion :**

Dans ce mémoire, nous avons souligné les exigences en matière de détection d'intrusion réseau. Nous avons présenté quelques systèmes existants et illustré leurs limites. Principalement, la flexibilité, l'autonomie, l'adaptabilité et la distribution ont été les principales caractéristiques à prendre en compte pour construire une architecture appropriée qui réponde à ces exigences. Ainsi, l'introduction d'un système multi-agents a été proposée comme moyen de modéliser et de mettre en œuvre une décision adaptative. Le système multi-agents rend la détection

d'intrusion plus flexible. En effet, l'autonomie accordée aux agents réduit considérablement l'implication du responsable de la sécurité dans la gestion de la sécurité et facilite ses tâches d'administration.

# *Chapitre 3 :*

## Contribution

## 1. Introduction :

Dans ce chapitre on proposer un système de détection d'intrusion basé sur un système multi-agent précisément les agents mobiles. Premièrement on présente notre architecture du système puis, le langage de programmation et l'environnement de travail et les outils utiliser pour mètre on œuvre cette IDS (Jade, Weka et eclipse), et concernant les données d'apprentissage on utilise les fichiers csv de NSL-KDD. Enfin, on expliquer le fonctionnement de notre système Et sa fiabilité dans la détection d'intrusion avec la démonstration de quelque mesure.

## 2. Motivations :

L'évolution des réseaux, en termes de nombre d'utilisateurs et de services, les rend toujours plus complexes et par conséquent vulnérables à de nouveaux types d'attaques. Les systèmes de détection d'intrusions existants ne sont pas facilement adaptables à cette complexité. Un autre facteur important est que le traitement des données d'une manière centralisée induit des vulnérabilités dans le système informatique, en l'occurrence, des réductions de performances en termes de scalabilité, configurabilité et de tolérance aux pannes. En effet, il suffit que l'engin central soit défectueux pour avoir une défection de tout le système de détection des intrusions. En outre, un grand flux d'événements peut entraîner un ralentissement du temps de traitement des données, une réaction tardive du système, une surcharge du réseau voire même des pertes de données rendant toute analyse biaisée. Notons aussi que la centralisation freine l'extensibilité du système et rend sa reconfiguration difficile.

Les agents mobiles nous permettent de faire le calcul distribué. L'idée est de faire transporter l'analyseur vers les flux d'audit et non les flux d'audit vers l'analyseur. Nous nous proposons de développer un modèle quant à l'utilisation de la technologie des agents mobiles Intégré avec la technologie de la machine learning dans la détection des intrusions.

## 3. Objectifs de l'approche :

L'approche que nous proposons a pour objectif de remplir les fonctions nécessaires à la détection des attaques de sécurité. Pour réaliser cela, notre solution doit fournir les avantages suivants :

- Une approche distribuée pour détecter les attaques qui se caractérisent par des signatures
- Un modèle utilisant la technologie d'agents mobiles pour distribuer la détection d'intrusions ;

- Un modèle utilisant la technologie de la machine Learning afin que chaque agent puisse détecter les différentes attaques et en déterminer leurs types, ce qui en fait un modèle plus fiable.

Dans ce qui suit on va présenter les détails de cette approche.

## 4. Le modèle proposé : une approche à base d'agents mobiles pour un IDS distribué

Dans cette section, nous allons présenter notre modèle de détection d'intrusion. Premièrement nous décrivons l'architecture générale **(Figure 12**), et nous détaillons les entités composantes de notre système de détection d'intrusion. Puis nous donnons quelques informations de nos agents et les objets manipulés par ces mêmes agents.

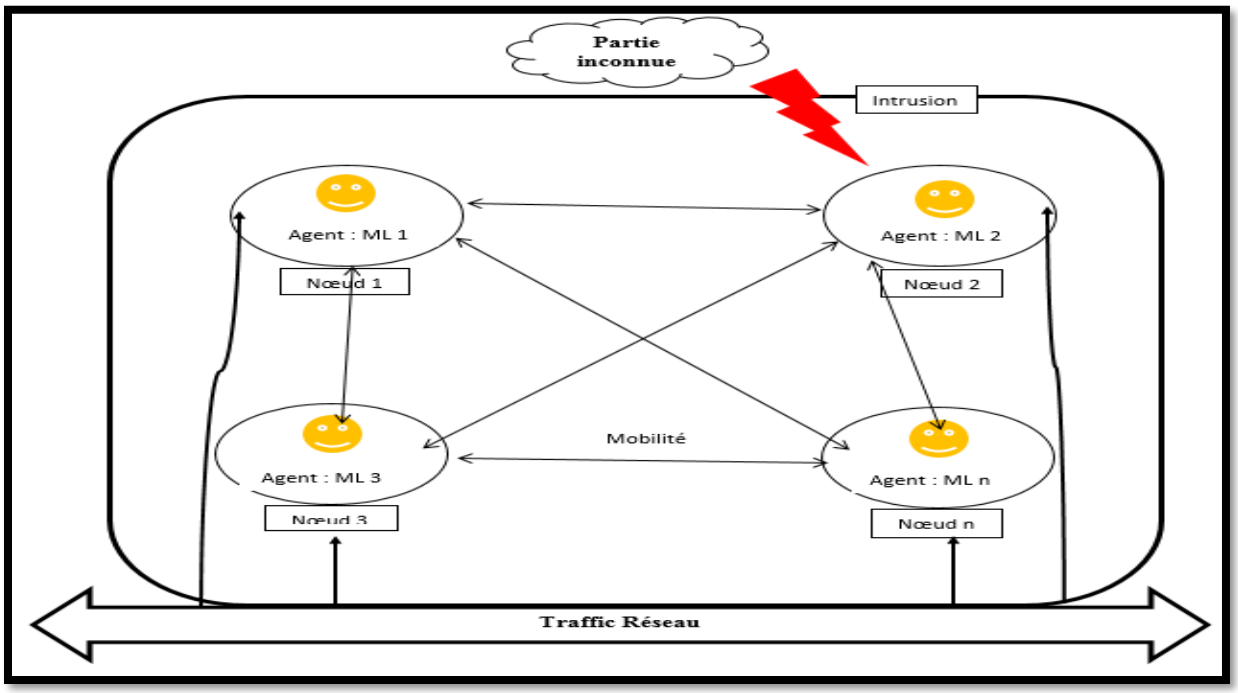



**Figure 13 :** Architecture générale du modèle proposé

Notre architecture se compose de plusieurs conteneurs, qui contiennent à leur tour un agent, doté de la fonctionnalité de la mobilité, ce qui signifie qu'il est possible de passer d'une machine à une autre, et qu'il s'agit d'une machine learning de sorte que chacun d'entre eux est différent et il se base sur des différent classifieur pour permettre l'analyse d'un nœud réseau dans lequel il se trouve et selon les résultats de l'analyse ils vont déterminer s'il y a une attaque ou non.

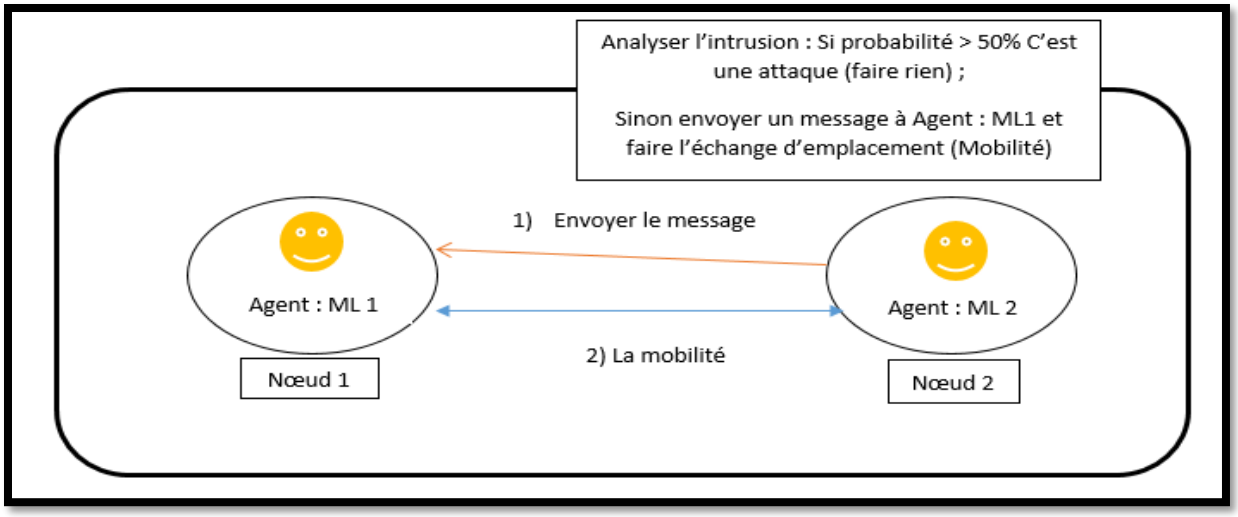



**Figure 14 :** condition de mobilité

La communication entre les agents se fait à partir de l'expédition et la réception des message ACL (Agent Communication Langage) (**Figure 13**), Afin d'envoyer le résultat de la probabilité et selon la condition les deux agents se déplacent vers leurs nœuds en échangeant les conteneurs (**Figure 14**).

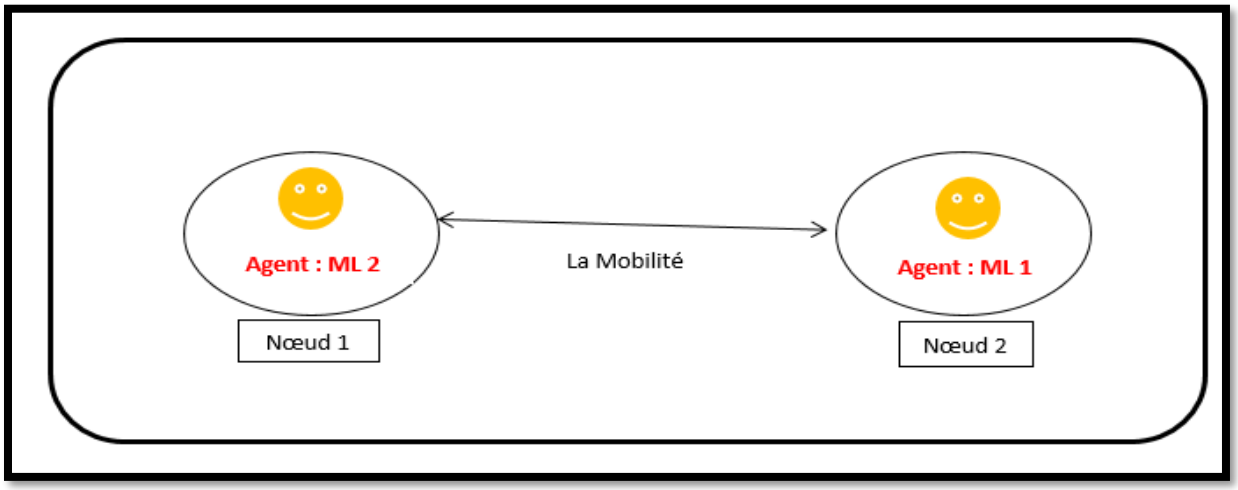



**Figure 15 :** Le changement d'emplacement

## 5. Outils de développement :

### Le langage Java :

Pour la mise en œuvre de notre système, nous allons choisir le langage de programmation orienté objet « Java » développé par Sun Microsystems. Ce langage a réussi à intéresser beaucoup de développeurs à travers le monde. En effet, Java est un langage multiplateforme disposant d'une machine virtuelle appelée JVM (Java Virtual Machine) lui permettant de

s'exécuter sur n'importe quelle machine. Java est capable de tourner aussi bien sur un PC que sur un MAC, sur un téléphone ou encore sur une carte à puce.

Pour programmer avec java dans notre application, nous utilisons la version 8 du JRE (Java Runtime Environment).

## L'IDE Eclipse Indigo :

est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

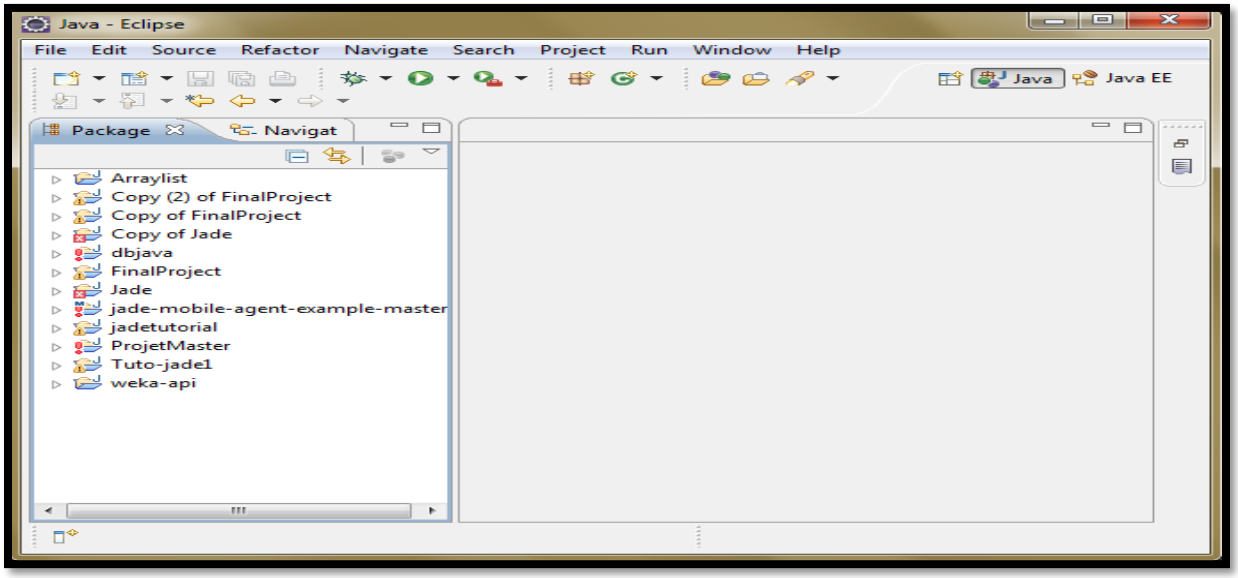



**Figure 16 :** Interface graphique de eclipse

## JADE (Java Agent Development Framework) :

est une plateforme de programmation multi-agent implémentée en Java. Les agents qui tournent sous JADE communiquent via le langage Agent Communication Language ou ACL.

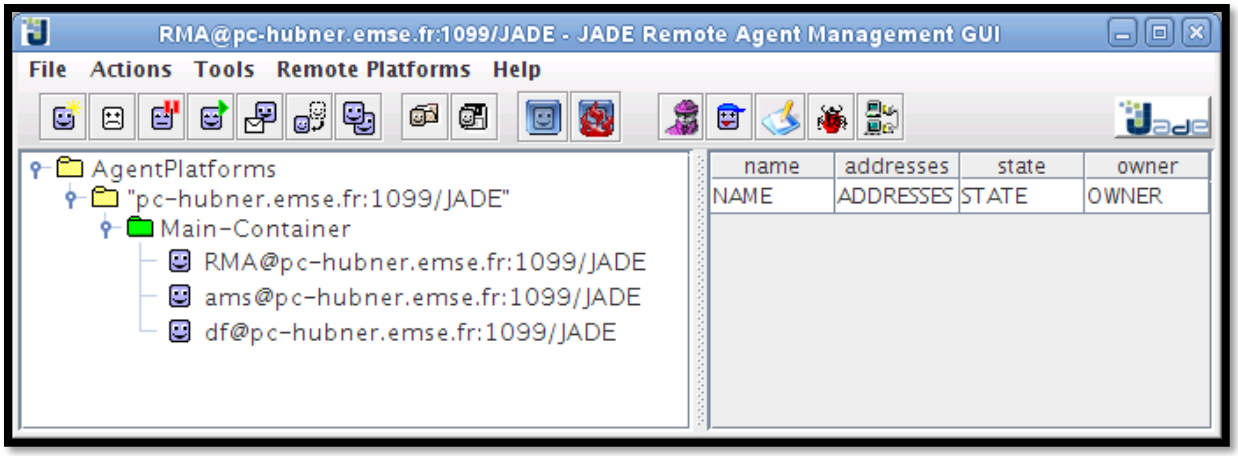



**Figure 17 :** Interface graphique de Jade

### Weka :

Est une suite de logiciels d'apprentissage automatique écrite en Java et développée à l'université de Waikato en Nouvelle-Zélande. Weka est un logiciel libre disponible sous la Licence publique générale GNU.

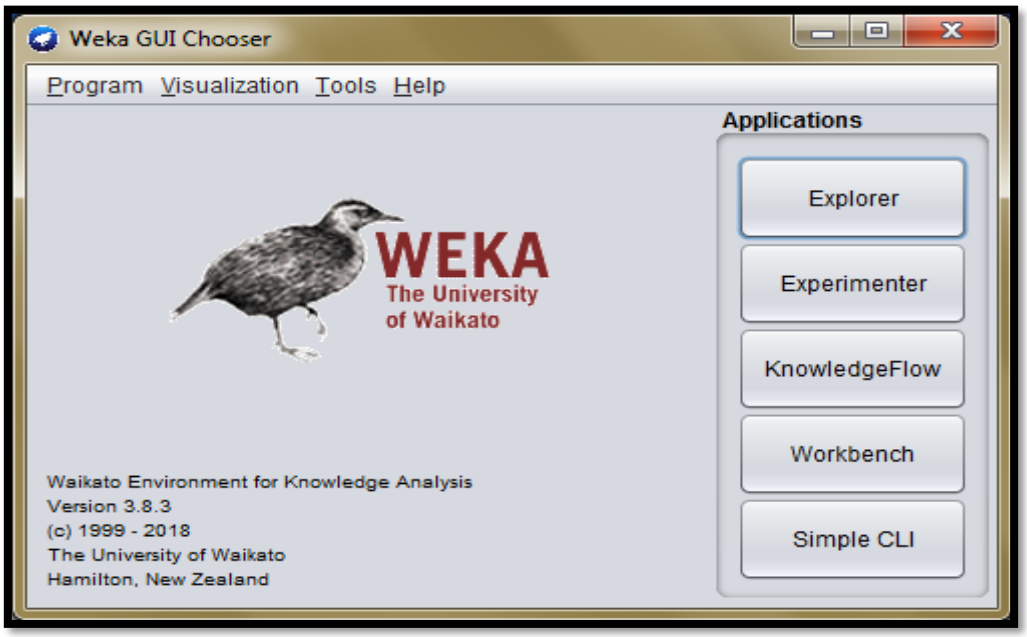



**Figure 18 :** Interface graphique de Weka

## 6. Le fonctionnement de notre modèle :

### 6.1. La phase d'apprentissage :

Dans cette phase, on forme notre modèle dans le but de le préparer pour la phase de test. On utilise quatre types de classifieur de la machine learning WEKA qui sont :

- **Naïve Bayes (Naïf Bayésienne):** Un classificateur naïf de Bayes est un algorithme qui utilise le théorème de Bayes pour classifier les objets. Les classificateurs naïfs de Bayes supposent une indépendance forte ou naïve entre les attributs des points de données. Parmi les utilisations courantes des classificateurs naïfs de Bayes figurent les filtres anti-spam, l'analyse de texte et les diagnostics médicaux. Ces classificateurs sont largement utilisés pour l'apprentissage automatique car ils sont simples à mettre en œuvre.

Naïve Bayes est également connu sous le nom de simple Bayes ou Bayes de l'indépendance **[60].**

- **Decision Tree (Arbre de Décision) :** Un arbre de décision est une représentation graphique de situations de décision spécifiques utilisées lorsque des ramifications complexes se produisent dans un processus de décision structuré. Un arbre de décision est un modèle prédictif basé sur une série de tests booléens de branchement utilisant des faits spécifiques pour tirer des conclusions plus générales [61].
- **Neural Network (Réseau de Neurones) :** Un réseau de neurones est une série d'algorithmes visant à reconnaître les relations sous-jacentes dans un ensemble de données via un processus reproduisant le fonctionnement du cerveau humain. Les réseaux de neurones peuvent s'adapter aux changements d'entrée; le réseau génère donc le meilleur résultat possible sans avoir à repenser les critères de sortie. Le concept de réseaux de neurones, qui trouve ses racines dans l'intelligence artificielle, gagne rapidement en popularité dans le développement des systèmes de trading [62].
- **Repeated Incremental Pruning to Produce Error Reduction (RIPPER) :** Cette classe implémente un apprenant à la règle propositionnelle, Élagage incrémentiel répété pour produire une réduction des erreurs (RIPPER), proposé par William W. Cohen en tant que version optimisée de IREP. Elle repose sur des règles d'association avec élagage réduit des erreurs (REP), technique très courante et efficace utilisée dans les algorithmes d'arbre de décision [63].

### 6.2. Les données d'apprentissage:

Dans cette phase, nous testons la performance de notre modèle après l'achèvement de la phase d'apprentissage. Seul l'ensemble de données de test est utilisé pour accomplir cette étape. L'ensemble de données qu'on utiliser est NSL-KDD Dataset

### 6.2.1. NSL-KDD Dataset :

NSL-KDD est un ensemble de données proposé pour résoudre certains des problèmes inhérents au KDD'99 dataset. Bien que cette nouvelle version d'ensembles de données KDD pose encore quelques problèmes et ne soit pas un représentant idéal des réseaux réels actuels, nous pensons qu'elle peut toujours être utilisée comme un dataset de référence efficace pour aider les chercheurs à comparer différentes méthodes de détection [64].

### 6.2.2. Les avantages de NSL-KDD :

Le NSL-KDD dataset présente les avantages suivants par rapport au KDD'99 dataset d'origine [64]:

- Il n'inclut pas les enregistrements redondants dans l'ensemble d'apprentissage, de sorte que les classificateurs ne seront pas orientés vers des enregistrements plus fréquents.
- Il n'y a aucun enregistrement en double dans les ensembles de tests proposés; par conséquent, les performances des apprenants ne sont pas biaisées par les méthodes qui ont de meilleurs taux de détection sur les enregistrements fréquents.
- Le nombre d'enregistrements dans l'ensembles d'apprentissage et les tests est raisonnable, ce qui permet de réaliser des expériences sur l'ensemble complet sans qu'il soit nécessaire de sélectionner au hasard une petite partie. Par conséquent, les résultats d'évaluation de différents travaux de recherche seront cohérents et comparables.

## 6.3. Pseudo code d'un agent mobile:

```
Début
Y = 50;
Start agent;
        Print "Hello Jade! I'm the X agent."
Charge dataset-Train;
Construire un classificateur
Charge dataset-Test;
    Pour i et le plus petite que datasetTest.numInstances() faire
        Prendre une Instance du dataset-Test
        Obtenir une prédiction de cette Instance
                Print "La probabilité du classe"
            Si b est plus petit que Y alors
                Envoyer un message au agent adjacent ;
            finsi
```

finpour

Si reçoive un message alors

Activer le comportement de la mobilité ;

Fin

## 7. L'implémentation du model :

Pour qu'on puisse utiliser les outils mentionnés précédemment il faut importer leur bibliothèque dans notre projet qui sont : « Weka.jar » et « Jade.jar »

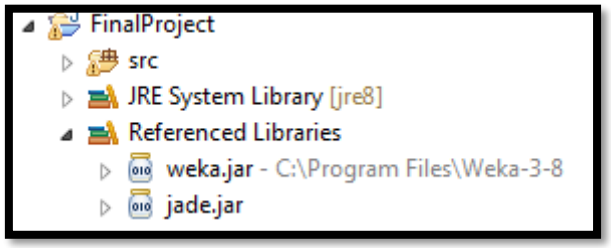



**Figure 19 :** les fichiers Jar utiliser

On utilise les fichiers de NSL-KDD normalisés :

- Pour l'apprentissage : normal_training20.csv ou la taille du fichier est 40000 instance.
- Pour le teste : le fichier normal_testing.csv ou la taille du fichier est 199996 instance.

Le programme qui implémente notre model contient l'ensembles des classes suivantes :

- **La classe Main :** c'est la classe responsable sur le lancement du Jade et à la création des conteneurs
- **La classe NBAgent :** la classe de l'agent qui utilise la méthode de classification Naïve Bayes.
- **La classe DTAgent :** la classe de l'agent qui utilise la méthode de classification Decision Tree.
- **La classe JRIPAgent :** la classe de l'agent qui utilise la méthode de classification RIPPER.
- **La classe NNAgent :** la classe de l'agent qui utilise la méthode de classification Neural Network.

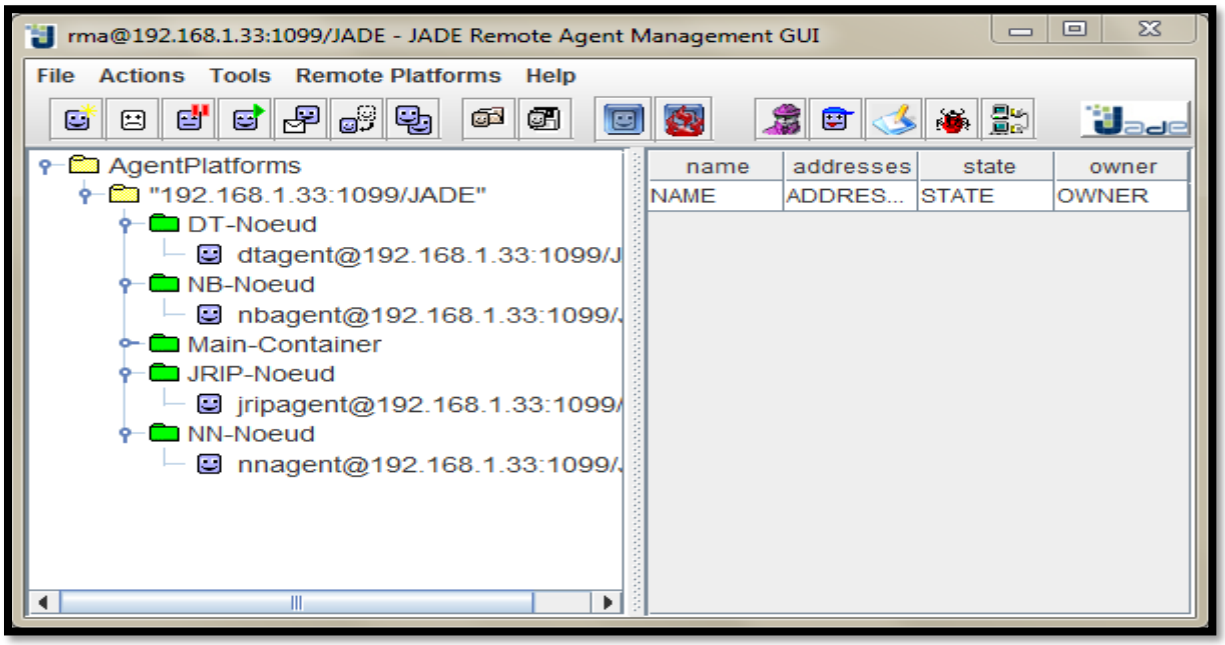



**Figure 20 :** Les agents de notre projet (avant la mobilité)

## 7.1. Un extrait de code :

```
package idsApp;


import idsApp.NNAgent.Mobilité;

public class NBAgent extends Agent {
    class counter{
        int mobilitycounter;
        public void mobilitycounter(){mobilitycounter++;}
    }
    private static final long serialVersionUID = 1L;


    @Override
    public void setup() {
        System.out.println("Hello Jade! I'm the NB  agent.");System.out.println("============================");
       try {
         //Read all the instances of dataset train in the file (ARFF, CSV, XRFF, ...)
            DataSource source = new DataSource("C:/Users/Bouthaina/Desktop/Mastére/master 2/Mémoire de fin d'étude/NSL-KDD DataSet/normal_training20.csv");
            Instances datasetTrain = source.getDataSet();//////////////////////

            //Make the last attribute be the class
            datasetTrain.setClassIndex(datasetTrain.numAttributes()-1);//////////////////////

            // categorie de class
        int numClasses = datasetTrain.numClasses();
        for (int i=0;i<numClasses;i++){
        String classValue = datasetTrain.classAttribute().value(i);
        System.out.println("the "+i+"th class value of NBAgent:"+classValue);}
        System.out.println("============================");

                //create the NB Classifier
              NaiveBayes nb = new NaiveBayes();
              nb.buildClassifier(datasetTrain);//////////////////

              //Read all the instances of dataset test in the file (ARFF, CSV, XRFF, ...)
              DataSource source2 = new DataSource("C:/Users/Bouthaina/Desktop/Mastére/master 2/Mémoire de fin d'étude/NSL-KDD DataSet/normal_testing.csv");
              Instances datasetTest = source2.getDataSet();
```

**Figure 21 :** La création d'un agent

```
//make prediction by Naive Bayes (NB) classifier
System.out.println("==============================");
System.out.println("Actual Class, NB Predicted");
double tp = 0.0, fp = 0.0, tn = 0.0, fn = 0.0, DR = 0.0, predNB = 0.0, FAR = 0.0, Accuracy = 0.0;
Evaluation evaluation = new Evaluation(datasetTrain);
evaluation.evaluateModel(nb, datasetTest);
tp = evaluation.numTruePositives((int) predNB);
fp = evaluation.numFalsePositives((int) predNB);
tn = evaluation.numTrueNegatives((int) predNB);
fn = evaluation.numFalseNegatives((int) predNB);
for (int i = 0; i < datasetTest.numInstances(); i++) {
    Instance newInst = datasetTest.instance(i);
    predNB = nb.classifyInstance(newInst);}
                    System.out.println("  True positives : "+" "+ tp);
                    System.out.println("  False positives: "+" "+ fp);
                    System.out.println("  True negatives : "+" "+ tn);
                    System.out.println("  False negatives: "+" "+ fn);
                    System.out.println("==============================");
                    System.out.println(evaluation.toMatrixString("Confusion matrix:"));


                    System.out.println("==============================");
                    DecimalFormat df = new DecimalFormat("0.00");
                    // Calculer DR
                    DR = ((tp/(tp+fn)))*100;
                    System.out.println("DR : "+" "+ df.format(DR) +" "+"%");
                    // Calculer FAR
                    FAR = ((fp/(fp+tn)))*100;
                    System.out.println("FAR : "+" "+df.format(FAR) +" "+"%");
                    // Calculer Accuracy
                    Accuracy = (((tp+tn)/(tp+tn+fp+fn)))*100;
                    System.out.println("Accuracy : "+" "+ df.format(Accuracy) +" "+"%");

                   System.out.println("==============================");
                   counter mc = new counter();

                   addBehaviour(new Mobilité(this) );
```

**Figure 22 :** L'évaluation du classifieur Naïve Bayes

```
public class Mobilité extends OneShotBehaviour{
      public Mobilité(Agent agent){}
      //create some varriables
      Random rand = new Random();
      ContainerID destination = new ContainerID();
          @Override
          public void action() {
              //create list of container name
              ArrayList<String> containerName = new ArrayList<String>();
              containerName.add("DT-Noeud");
              containerName.add("JRIP-Noeud");
              containerName.add("NN-Noeud");

              //choose randomly name of container
              String contNamedest= containerName.get(rand.nextInt(containerName.size()));

              //send it to other agents
              ACLMessage msgS=new ACLMessage(ACLMessage.INFORM);
              msgS.setContent(contNamedest);
              msgS.addReceiver(new AID("dtagent",AID.ISLOCALNAME));
              msgS.addReceiver(new AID("jripagent",AID.ISLOCALNAME));
              msgS.addReceiver(new AID("nnagent",AID.ISLOCALNAME));
              send(msgS);//

              //do the action
              System.out.println("container Name deststination"+" "+ contNamedest);
              destination.setName(contNamedest);
              myAgent.doMove(destination);}}
          protected void beforeMove() {
                try {System.out.println("Moving from location : " + getContainerController().getContainerName());
              } catch (ControllerException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();}}
             protected void afterMove() {
                 try {System.out.println("Arrived at location : " + getContainerController().getContainerName());
              } catch (ControllerException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();}}}
```

**Figure 23 :** La mobilité de l'agent

## 7.2. Les résultats :

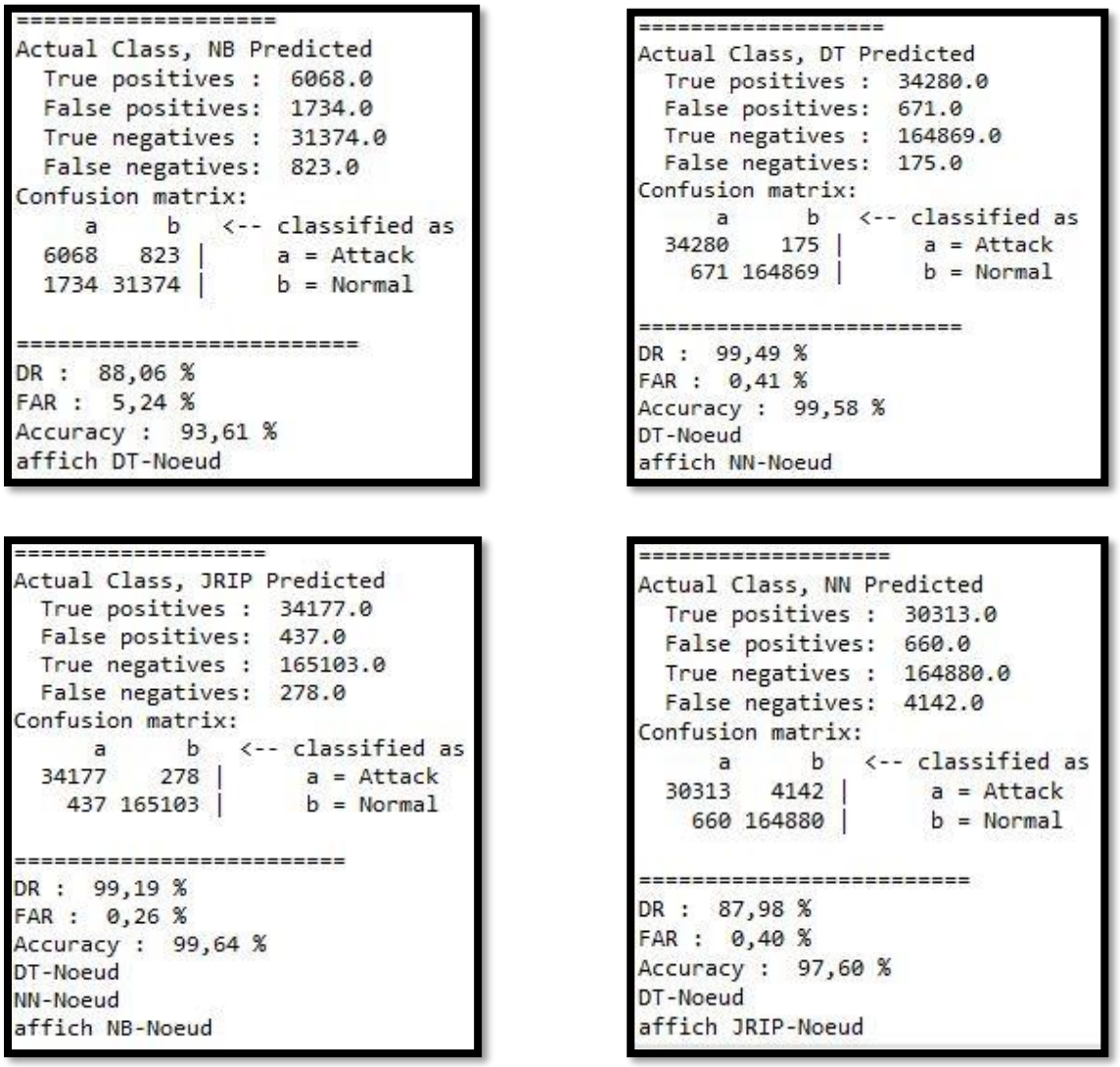



**Figure 24 :** Les résultats de chaque agent de notre système

La **Figure 22** illustre les résultats de notre système concernant chaque agent par exemple si on prend les résultats de l'agent DTAgent (Decision Tree) on voit qu'il est le meilleur agent ; 34280 attaques (True Positive) et 164869 normale (True Négative) et pour les attaques qui n'arrive pas à détecter 671 attaques (False Positive) et pour les False Négative juste 175 attaques, c'est pour ça il a les meilleurs résultats de performance par une DR égale à 99.49 % et FAR égale à 0.41 %, Accuracy égale à 99.58 %.

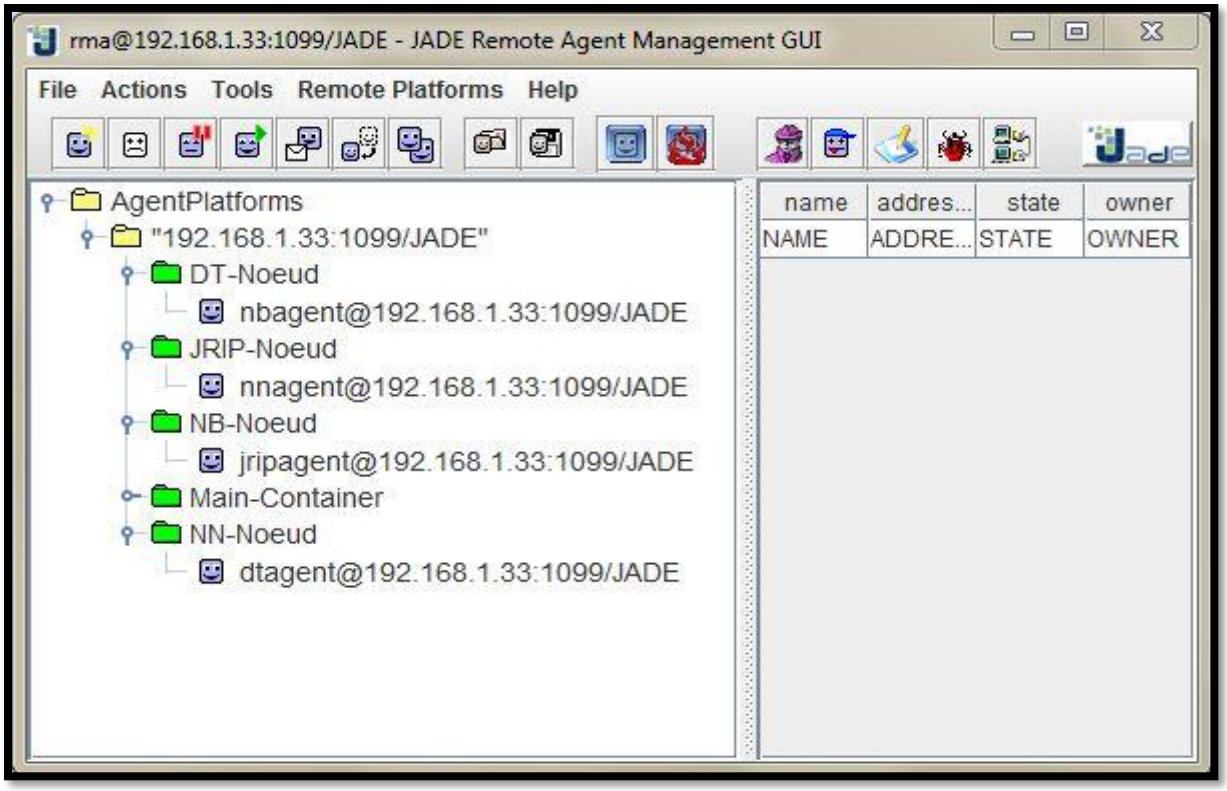
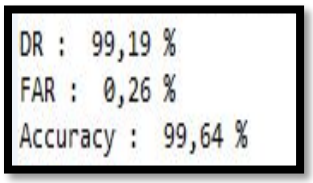



**Figure 25 :** Les agents après la mobilité

La **Figure 23** illustre la mobilité des agents de système, Chaque agent analysera le nœud de réseau où il se trouve après la mobilité et obtiendra les résultats de la performance. Par conséquent, nous calculerons la moyenne entre les résultats des deux agents pour voir l'efficacité de deux agents dans un seul nœud.

- Par exemple on prend le JRIP-Nœud :

On calcul la moyenne entre les résultats du « jripagent » et « nnagent »



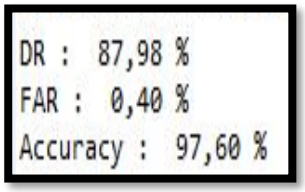

Jripagent

DR = 93.585 % et

FAR = 0.33 % et

Accuracy = 98.62 %

La moyenne




Nnagent

## 8. Conclusion :

Un système de détection d'intrusion (IDS) est un système qui surveille le trafic réseau à la recherche d'activités suspectes, dans ce contexte on a développé un système de détection d'intrusion distribué basé sur l'agent mobile et qu'il se base sur leur apprentissage sur des

différents types de classifieur pour améliorer la fiabilité du notre système. On a obtenu des bons résultats de performance par une DR égale à 93,68 % et FAR égale à 1,5775 %, Accuracy égale à 97,6075 %.

# *Conclusion Générale*

**Conclusion générale :**

Le travail présenté dans ce mémoire s'inclut dans le domaine de la sécurité informatique et précisément les systèmes de détection d'intrusion.

On a proposé un modèle pour la détection d'intrusion distribué qu'il se base sur les systèmes multi-agents précisément les agents mobiles. Ce modèle est proposé pour résoudre quelques problèmes des anciens systèmes qu'ils ont la même architecture distribuée.

Les Agents mobiles de notre système effectuent une détection d'anomalie avec les signatures des attaques. Durant la phase d'apprentissage chaque agent va utiliser une méthode de classification différente a l'autre qui sont « Naïve Bayes », « Decision Tree », « RIPPER algorithme », « Neural Network » et l'apprentissage automatique se fait à base de NSL-KDD dataset qui contient un ensemble des signatures des attaques pour deux catégories « Attack » et « Normal ».

On a fixé un seuil *s*=50% comme un minimum de confiance, si la valeur du test dépasse *s* alors l'agent traite l'enregistrement comme « normal » sinon il choisit un agent aléatoirement pout permuter les nœuds.

**Perspective :**

Nous comptons poursuivre ce travail de la manière suivante :

- Amélioré ce travail en utilisant une nouvelle dataset d'actualité et qui contient les quatre catégories d'attaque classifier par DARPA (DoS, Probing, R2L, U2R), sous forme d'une base de données pour faciliter la manipulation des signatures des attaques.
- Compléter ce travail pour avoir une application fiable et commerciale.

# *Bibliographie*

# Bibliographie

[1] Cédric Michel. Langage de description d'attaques pour la détection d'intrusions par corrélation d'événements ou d'alertes en environnement réseau hétérogène. Informatique. Université Rennes. 2003

[2] Réseau CERTA. Système d'information: Qu'est-ce qu'un système d'information ?. Décembre 2005

[3] Becquet, V. Le programme national : Sécurité des systèmes d'information. 2003

[4] S. Ghernaouti-Héli, Stratégie et ingénierie de la sécurité des réseaux. Inter Editions, 1998.

[5] S. Ghernaouti-Héli, Sécurité Internet Stratégies et technologies. Dunod, 2000.

[6] M Thibaut Probst : évaluation et analyse des mécanismes de sécurité des réseaux dans les infrastructures virtuelles de cloud computing,2015.

[7] Mme LABED Ines : Proposition d'un système immunitaire artificiel pour la détection d'intrusions, université de Constantine, 2006.

[8] ISO/IEC 27000. (2009), Information technology — Security techniques — Information security management systems — Overview and vocabulary, http://standards.iso.org/ittf/licence.html

[9] Cole, E., Krutz, R., Conley, J. (2005), Network Security Bible, Wiley Publishing, Inc, and ISBN13:978-0-7645-7397-2.

[10] Przemysiam Kazienko & Piotr 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

[11] William Stallings, network security essentials: applications and standards fourth edition, 2011.

[12] Aissaoui Sihem, Apprentissage automatique et sécurité des systèmes d'information : Application un système de détection d'intrusion basé sur les (SVM), Université d'oran,2008.

[13] Anderson J, Computer security threat monitoring and surveillance, 1980.

[14] D. E. Denning, An intrusion detection model, in IEEE Transactions on software engineering, SE-13 :222-232, 1987.

[15] Philippe Biondi, Architecture expérimentale pour la détection d'intrusions dans un système informatique,2001.

[16] Endorf, C., Schultz, E., Mellander, J, Intrusion Detection and Prevention, ISBN: 0072229543, 2004.

[17] L. Mé et C. Michel, La détection d'intrusions : bref aperçu et derniers développements. Actes du congrès EUROSEC'99, 1999.

[18] Debar, H., Dacier, M., Wespi, A, A Revised Taxonomy for Intrusion-Detection Systems, Annales des Télécommunications, Vol. 55, No. 7-8, pp. 361-378, 2000.

[19] Vinod Kumar, Dr. Om Prakash Sangwan, "Signature based intrusion detection system using Snort" International Journal of Computer Applications & Information Technology Vol. I, Issue III, November 2012 (ISSN: 2278-7720)

[20] Guan Xin and Li Yun-jie, "A new Intrusion Prevention Attack System Model based on Immune Principle", International Conference on e-Business and Information System Security (EBISS), in IEEE, pp. 1-4, 2010.

[21] G. Vigna, S. Eckmann and R. Kemmerer, Attack Languages, in Proceedings of the IEEE Information Survivability Work-shop, USA, pp. 163-166, 2000.

[22] Cédric Michel, Langage de description d'attaques pour la détection d'intrusions par corrélation d'événements ou d'alertes en environnement réseau hétérogène, Université de Rennes 1,2003.

[23] A. Pharate, H. Bhat, V. Shilimkar, N. Mhetre, "Classification of Intrusion Detection System", International Journal of Computer Applications (0975 – 8887), Volume 118 – No. 7, May 2015

[24] Yousef Farhaoui, « Evaluation des systèmes de détection et de prévention des intrusions et la conception d'un BIDS », thèse de doctorat, Université Ibn Zohr, 2012.

[25] J.Allen, A.Christie «State of the practice of intrusion detection technologies ». Publisher in proceeding of Networked Systems Survivability Program CMU/SEI-99TR-028, 2000.

[26] Vasilomanolakis, Emmanouil & Karuppayah, Shankar & Mühlhäuser, Max & Fischer, Mathias. Taxonomy and Survey of Collaborative Intrusion Detection. ACM Computing Surveys. 47. 10.1145/2716260, 2015.

[27] G. Price Mukherjee, L.Heberlein «Network Intrusion Detection » publisher in IEEE Network vol 3 (3) pp 26-4, 1994,

[28] Porras. P. A & Neumann. P. G. « EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances », Proceeding of 20th National Information System Security Conference, 1998.

[29] J. S. Balasubramaniyan & J. O. Garcia-Fernadez & D. Isacoff & E. H. Spafford & D. Zamboni. « An Architecture for Intrusion Detection using Autonomous Agents ». Technical Report Coast-TR-98-05, Computer Sciences Department, Purdue University, 1998.

[30] J. McCarthy, Marvin L. Minsky, N. Rochester, and Claude E. Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence",(1956), AI Magazine Volume 27 Number 4 (2006).

[31] Alsedrah, Mariam. Artificial Intelligence. 10.13140/RG.2.2.18789.65769, 2017.

[32] Ferber. « L'intelligence Artificielle Distribuée ». Publisher in La Recherche, 1991, vol 23(3), pp 750-758.

[33] Alrajeh, Nabil & Lloret, Jaime. Intrusion Detection Systems Based on Artificial Intelligence Techniques in Wireless Sensor Networks. International Journal of Distributed Sensor Networks. 2013.

[34] H. Om, T. Hazra, "STATISTICAL TECHNIQUES IN ANOMALY INTRUSION DETECTION SYSTEM", International Journal of Advances in Engineering & Technology, Nov. 2012.

[35] N. Bashah Idris, B. Shanmugam "Artificial Intelligence Techniques Applied to Intrusion Detection", IEEE Indicon 2005 Conference, Chennai, India, 11 - 1 3 Dec. 2005.

[36] D. Boughaci, H. Drias, «Distributed intrusion detection frame work based on autonomous and mobile agents». In Proceedings of the International Conference on Dependability of Computer Systems, pp 248–255, 2006.

[37] Benyettou, Noria & Benyettou, Abdelkader & Rodin, Vincent. an Immune Agents System for Network Intrusions Detection. Computer Science & Information Technology, 2014.

[38] J.Briot, Y Demazeau, «Introduction aux agents : Principes et architecture des systèmes multi-agents», Book Collection IC2, Hermès, 2001.

[39] Wooldridge, M., & Jennings, N. *Intelligent agents: Theory and practice.* The Knowledge Engineering Review, 1995.

[40] Jarras, I., & Chaib-draa, B. Aperçu sur les systèmes multiagents - Série scientifique. Montréal: CIRANO, 2002.

[41] J. Ferber, «Les Systèmes Multi-Agents : Vers une Intelligence Collective», InterEditions, ISBN 2-7296-0665-3. 1997.

[42] H. Labiod, "Error Control in Wireless ATM networks", Ph.D thesis, Universityof Versailles, France, 1998.

[43] M.Cossentino, C. Potts. «A CASE tool supported methodology for the design of multi-agent systems», in Proceeding. The International Conference on Software Engineering Research and Practice (SERP'02) Las Vegas (NV), USA, 2002.

[44] Abdelhalim Zaidi. Recherche et détection des patterns d'attaques dans les réseaux IP à hauts débits. Networking and Internet Architecture [cs.NI]. Université d'Evry-Val d'Essonne, 2011. French. <Tel-00878783>

[45] Behrouz H. Far, "Co-ordination in Multi-Agent Systems: An Overview", SENG 609.22 – Agent-Based Software Engineering Tutorial report December, 2002

[46] A.Drogoul, «Systèmes multi-agents», Projet MIRIAD, Rapport technique, OASIS/LIP6, Université Paris 6, 2005.

[47] http://www.mind.ilstu.edu/curriculum/ants_nasa/intelligent_agents.php

[48] Je_rey M. Bradshaw. An introduction to software agents. In Je_rey M. Bradshaw, editor, Software Agents, chapter 1, pages 3{46. AAAI Press/The MIT Press, 1997.

[49] Perret, S. Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure, Thèse de doctorat de l'Université Joseph Fourier - Grenoble I, 1997.

[50] Jai Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, E. H. Spafford, and Diego Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents," Department of Computer Sciences, Purdue University; Coast TR 98-05, 1998.

[51] Frincke, D., Don Tobin, Jesse McConnell, Jamie Marconi, Dean Polla, "A Framework for Cooperative Intrusion Detection," Proceedings of the 21st National Information Systems Security Conference, pp. 361-373, October 1998.

[52] B. Clifford Neuman and Theodore Ts'o. "Kerberos: An Authentication Service for Computer Networks, " IEEE Communications, 32 (9), pp. 33 38, September 1994.

[53] W. Lee, S.J. Stolfo, and K. Mok, "A Data Mining Framework for Building Intrusion Detection Models," Proceedings of the IEEE Symposium on Security and Privacy, 1999.

[54] Guy Helmer, Johnny S. K. Wong, Vasant Honavar, and Les Miller.

"Intelligent Agents for Intrusion Detection." Proceedings, IEEE Information Technology Conference, Syracuse, NY, pp. 121-124, September 1998.

[55] M.Asaka, S.Okazawa, A.Taguchi, and S.Goto, "A Method of Tracing Intruders by Use of Mobile Agents," INET'99, June 1999.

[56] Bass,T. Multisensor data fusion for next gen-eration distributed intrusion detection systems.Pro-ceedings of the 1999 IRIS National Symposium on Sen-sor and Data Fusion.Symposium conducted at TheJohns Hopkins University Applied Physics Laboratory, 1999.

[57] Gomez, J., & Dasgupta D. Evolving fuzzyclassifiers for intrusion detection.Proceedings of the 3rdAnnual IEEE Information Assurance Workshop,NewOrleans, Lousiana, June 17–19, 2002.

[58] Jansen, W. Intrusion detection with mobile agents (2002,September). Computer Communications 25(15), Spe-cial Issue on Intrusion Detection, Elsevier, pp. 1392–1401, September 2002.

[59] AH. Boudjelida, *Réseaux Bayésiens Naïfs Augmentés TAN pour les Systèmes de Détection d'Intrusions*. Mémoire de Magistère ISI, 2008.

**[60]** **https://www.techopedia.com/definition/32335/naive-bayes**

**[61]** **https://www.techopedia.com/definition/28634/decision-tree**

**[62]** **https://www.investopedia.com/terms/n/neuralnetwork.asp**

**[63]** **https://www.revolvy.com/page/Repeated-incremental-pruning-to-produce-error-reduction-%28RIPPER%29**

**[64]** **https://www.unb.ca/cic/datasets/nsl.html**