Défense d'un réseau: couplage SMA et Système Immunitaire Artificiel

Mathis Baubriaud et Juliette Grosset

16 janvier 2021

Table des matières

1 Présentation du projet

Proposition d'une implémentation agent d'un système de détection d'intrusion basé AIS. Inspiré par la théorie immunitaire et les systèmes multi-agents, un modèle de défense immunitaire multi-agents actif pour l'intrusion de réseau est établi. Nous utiliserons pour la simulation le langage C++ et une bibliothèque fournie par Mr V.Rodin nous permettant d'instancier des agents et d'implémenter leurs interactions.

2 Pourquoi une approche multi-agents ?

Aujourd'hui, les technologies de détection d'intrusion, incluant l'analyse statistique, l'analyse des fonctionnalités, l'exploration de données, etc., ont toutes certaines limites. Ces techniques manquent d'auto-adaptation et ne peuvent détecter que les attaques connues. Ces systèmes manquent également de robustesse, ce qui signifie qu'une erreur dans un endroit local peut affecter l'ensemble des travaux.

Le système immunitaire biologique peut identifier et tuer les antigènes d'intrusion. Il peut être considéré comme un système autonome distribué. Avec la coopération et la coordination d'agents isolés dans un système multi-agents (SMA), les problèmes d'un environnement complexe peuvent être résolus et le système ainsi créé possède des caractéristiques de distribution et de robustesse.

Nous faisons dans cette étude, le lien entre les cellules immunitaires et les agents. Dans cette optique, le système immunitaire biologique est considéré comme un SMA. Nous utilisons donc la théorie immunitaire dans le système de défense actif multi-agent pour l'intrusion.

Système immunitaire biologique	Système de défense réseau
Organisme	Réseau dans la globalité
Organes	Section du réseau
Cellules	Hôtes
Vaccin	Information d'intrusion
Antigène	Texte extrait de paquets IP
Cellules B et T, anticorps	Anticorps exprimés par du texte
Cellule clone	La copie d'un anticorps

FIGURE 1 – Tableau des similitudes entre systèmes

Le but est d'établir la relation entre un système de détection d'intrusion dans un réseau et l'organisation de la défense du corps humain afin de simuler un mécanisme de sécurité actif.

3 Choix de modélisation

La simulation se fera avec le langage C++. C'est un langage orienté objet parfaitement adapté à la simulation multi-agents. La bibliothèque fournie par Mr V.Rodin nous a permis d'instancier des agents et des objets 2D ainsi que de définir les différentes interactions qui régissent l'environnement.

Du point de vue du système de défense réseau, l'environnement est représentée sur la figure 2. L'agent de surveillance local (LMA pour Local Monitor Agent) analyse l'état du réseau local, tandis que l'agent de surveillance central (CMA pour Central Monitor Agent) surveille l'ensemble du réseau. L'IA est le noyau de ce modèle, distribuant dans chaque nœud hôte les informations et identifiant les intrusions. Les IAs évaluent de manière active l'état de risque auquel le nœud est confronté et envoient l'état à l'agent de surveillance local (LMA) de son segment de réseau. Le LMA se charge de surveiller

le segment de réseau, de traiter les données de chaque IA, d'évaluer l'état de risque de son propre segment de réseau et de les envoyer à l'agent de surveillance central (CMA). Le LMA qui subit une intrusion envoie le vaccin aux autres segments sans intrusion pour les avertir. Ainsi, la défense active de l'ensemble du réseau est mise en œuvre. CMA prend en charge l'interaction homme-machine. Il reçoit l'état de risque du segment de réseau envoyé par LMA et montre l'état de sécurité via l'interface client. Ensuite, l'administrateur pourrait prendre des mesures de réponse pour protéger l'ensemble du réseau en temps voulu.

Dans la suite de cette étude, nous avons décrit les agents avec leurs représentations dans le système immunitaire biologique.

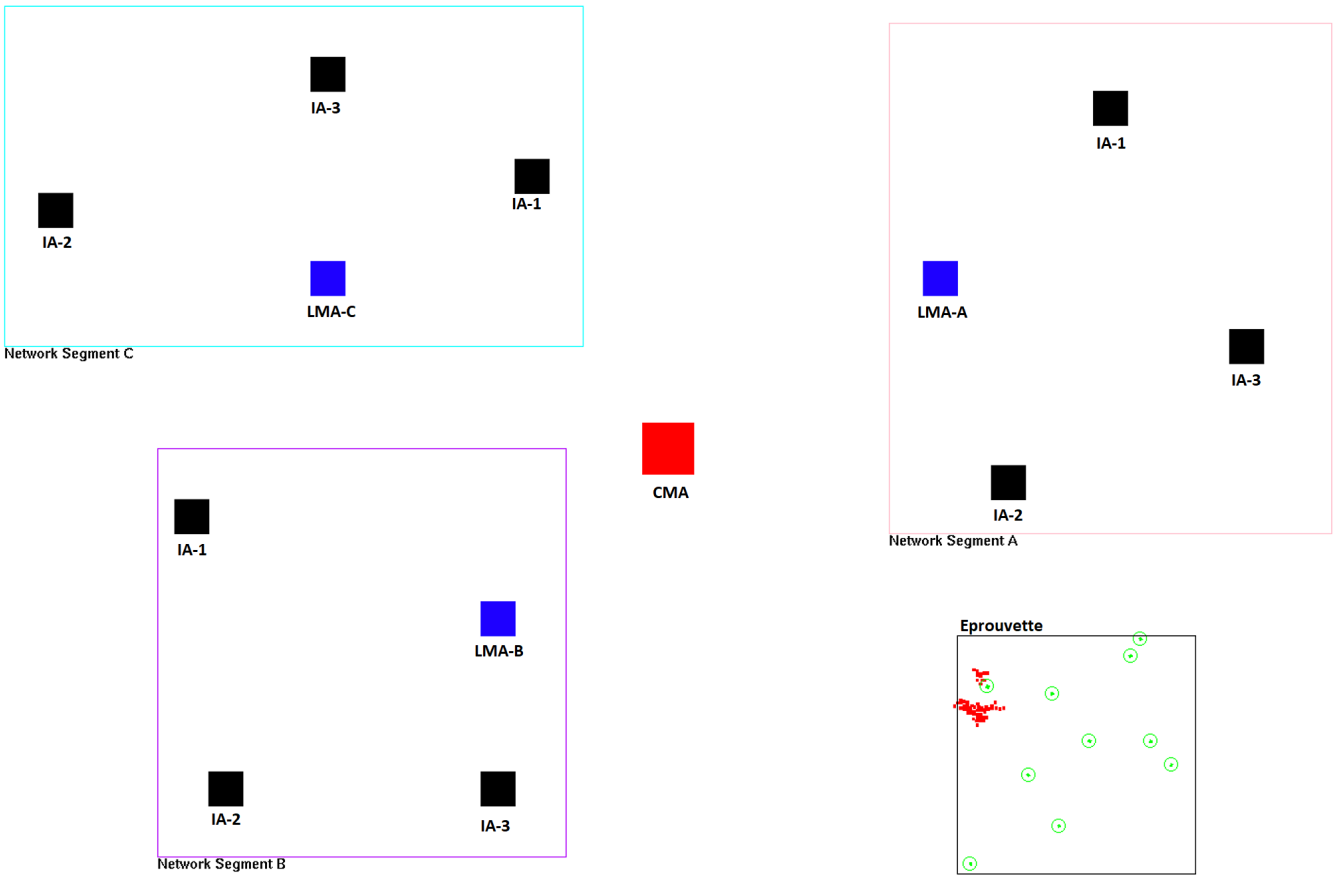

FIGURE 2 – Architecture de l'environnement de simulation

4 Les agents

Pour mieux différencier les agents, nous avons établi un code couleur. L'agent 2D rouge représente l'organisme, il est au centre de l'environnement. Les agents bleus sont les organes, compris dans une section délimitée par un rectangle. Les organes peuvent communiquer avec l'organisme. En noir, ce sont les cellules composant l'organe, qui sont en lien direct avec celui-ci. À l'intérieur des cellules se trouvent des anticorps, ceux-ci sont symbolisés dans l'encadré en bas à droite de l'environnement, censé évoquer une éprouvette. Les anticorps peuvent avoir 3 états : si aucun virus n'a été détecté dans la cellule, alors les anticorps sont immatures, ils sont représentés en vert dans la simulation. Ces anticorps ont un champ de perception et si un virus ou antigène, représenté en rouge, entre dans ce champ, alors les anticorps deviennent matures, leurs couleurs passent aux jaunes. Enfin, une fois le virus détecté, les anticorps devenus matures vont se déplacer vers celui-ci afin d'éliminer la menace. Si un anticorps phagocyte un virus, il retient le mécanisme de défense (vaccin) et devient mémoire. Pour récapituler, la simulation est composée de ses différents agents :

— Organisme : Agent rouge
— Organes : Agents bleus
— Cellules : Agents noirs
— Anticorps immatures : agents verts
— Anticorps matures : agents jaunes
— Anticorps mémoires : agents oranges
— Virus ou antigène : agents rouges.

5 Perception et Actions

Au démarrage de la simulation, on retrouve 1 organisme au centre (CMA) et 3 sections contenant chacunes 3 organes (LMA) et les cellules qui le compose (IA), au nombre de 3 par organes. (cf Fig.1). Dans chaque cellules se trouve 10 anticorps immatures. Un antigène va attaquer l'un des agents IA aléatoirement. L'antigène ou virus va se multiplier rapidement en se dupliquant, créant un nouvel antigène à une position adjacente à la sienne.

L'agent IA qui a été infecté va réagir à l'aide de ses anticorps. Ceux-ci disposent d'un champ de perception et, lorsqu'un virus est détecté, deviennent des anticorps matures et se déplacent vers le virus. Lorsqu'un virus est éliminé, l'anticorps devient mémoire, lorsqu'il n'y a plus de menace, il arrête de se déplacer.

Les agents IA et les agents LMA sont en communications permanentes. L'agent IA qui est attaqué par un antigène envoie un message au muscle de son secteur. Ce message contient l'avertissement qu'une menace est entrée dans le secteur ainsi que le vaccin si des anticorps mémoires sont présents dans l'agent IA.

L'agent LMA envoie deux types de signaux. L'un est destiné à l'organisme (CMA) pour l'avertir à tout moment du risque présent en son sein. L'autre est à destination des autres agents LMA et contient le vaccin qui lui aura été transmis par les agents IA qui auraient été infectés. De cette façon, dès qu'un virus est combattu, peu importe dans quels secteurs il est apparu, tous les autres secteurs vont pouvoir créer des agents IA mémoires qui seront capables de reconnaître l'antigène immédiatement s'il venait à apparaître. On peut voir l'architecture de subsumption d'un agent LMA dans la figure 7.

La capacité de défense de l'organisme est grandement améliorée de part ces interactions. L'organisme (CMA) fait office d'interface avec l'utilisateur extérieur. Il affiche en permanence l'état du risque présent grâce aux messages qu'il reçoit des agents LMA.

Interactions	Agent CMA	Agent LMA	Agent IA	Agent Ag	Agent Ac
Agent CMA	-	-	-	-	-
Agent LMA	risk status	-	-	-	-
Agent IA	-	detection d'attaque	-	-	-
Agent Ag	-	-	-	vaccin	élimine
Agent Ac	-	-	envahi	-	-

FIGURE 3 – Tableau des interactions

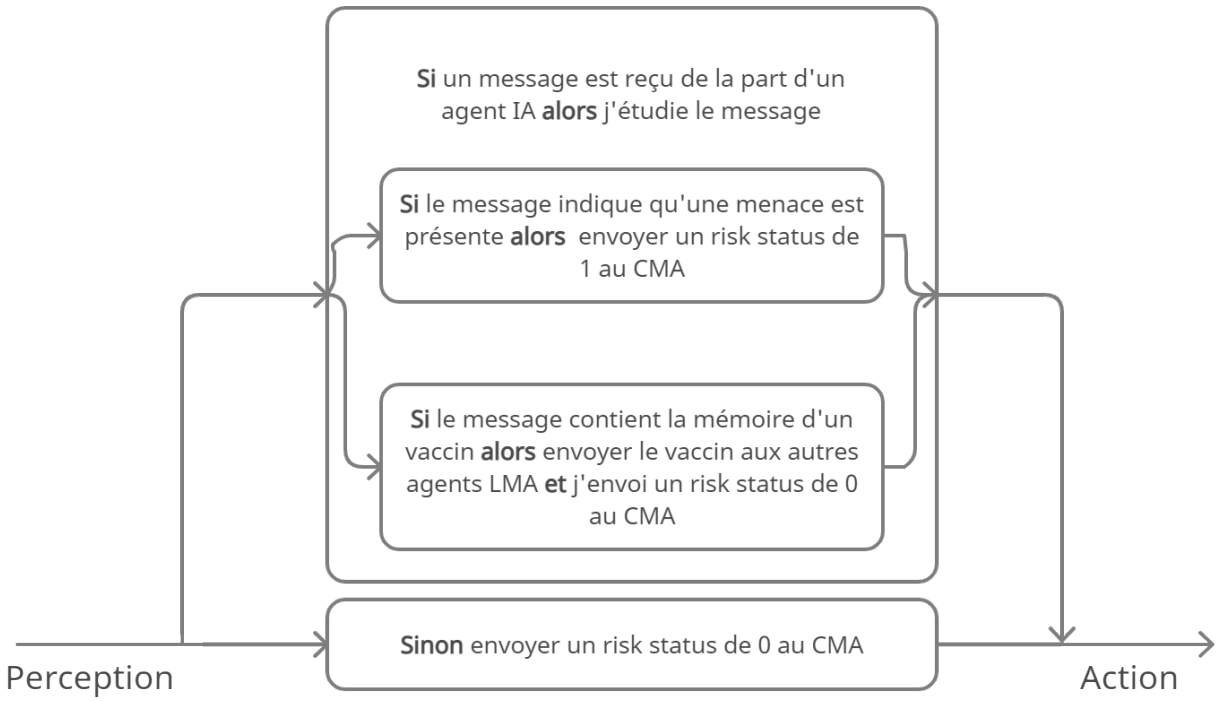

FIGURE 4 – Architecture de subsumbtion d'un agent LMA

6 Rôle de l'utilisateur

L'utilisateur a la possibilité d'interagir dans la simulation de différentes manières. Pour cela, il dispose de l'appui sur différentes touches possibles qui sont détaillées dans la console au lancement de la simulation.

```
*******************************************
help
----
Actions clavier AVEC objet selectionne:
- r sur CMA : risk status global
- h sur eprouvette : help de l'eprouvette
- n : obtenir le nom d'un Agent2D
- p : obtenir le nom de l'environnement d'un Agent2D
- v : visualiser la défense (éprouvette) de l'agent2D
Actions clavier SANS objet selectionne:
- h : help
- a : autoscale (oui/non)
-' ': pause     (oui/non)
- q : quitter
*******************************************
```

FIGURE 5 – Affichage du menu Help

Par exemple, à l'aide des touches n et p sur un Agent2D, il peut obtenir des informations sur celui, comme son nom, son environnement.

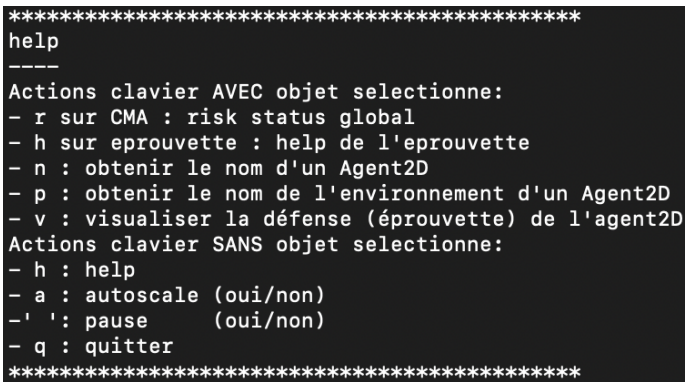

FIGURE 6 – Exemple d'affichage d'informations sur des Agents2D

L'appui sur la touche "V" sur un AgentIA, nous permet de visualiser sa défense. Celle-ci va alors s'afficher dans l'éprouvette en bas à droite de la fenêtre de visualisation. Le code couleur respecté est celui expliqué plus haut dans la section 4 concernant les agents. De plus, l'utilisateur peut avoir un rôle très important et interagir de manière directe sur cette simulation de cyberattaques de ce réseau. En effet, à l'aide de la touche H sur l'éprouvette, il peut voir les actions possibles dont il dispose.

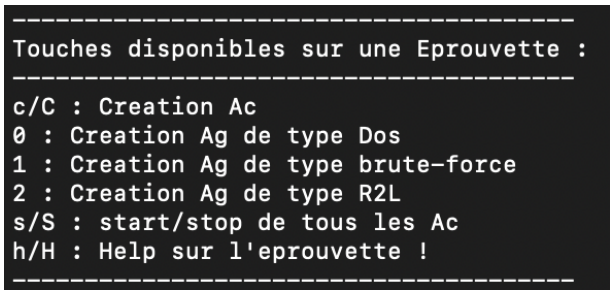

FIGURE 7 – Affichage du menu Help concernant l'éprouvette

Ainsi, on peut voir que l'utilisateur peut interagir et avoir une influence directe sur l'attaque du réseau en pouvant créer des Ag (virus) de différents types au sein d'un AgentIA. De manière similaire, il peut également aider un AgentIA à se défendre en créant des Ac.

7 Résultats

Nous avons réussi à correctement modéliser l'environnement. Un agent antigène Ag est placé dans un des agents IA et l'utilisateur est informé via un message dans la console. Ensuite, à l'aide de l'éprouvette, il peut se rendre sur la visualisation de l'agent en question pour observer la défense en direct des anticorps sur les antigènes. Des messages sont envoyés entre les agents pour alerter l'utilisateur sur le risque global présent dans le système. L'utilisateur peut ensuite regarder comment réagit le système lors de l'apparition de nouveaux agents antigènes de types différents (à l'aide des touches 0, 1 et 2). À tout moment, l'utilisateur peut également cliquer sur l'agent CMA central et appuyer sur la touche R pour avoir l'état du risque global actuel (0 = aucun risque, 1 = un antigène détecté, 2 = deux antigènes, etc..)

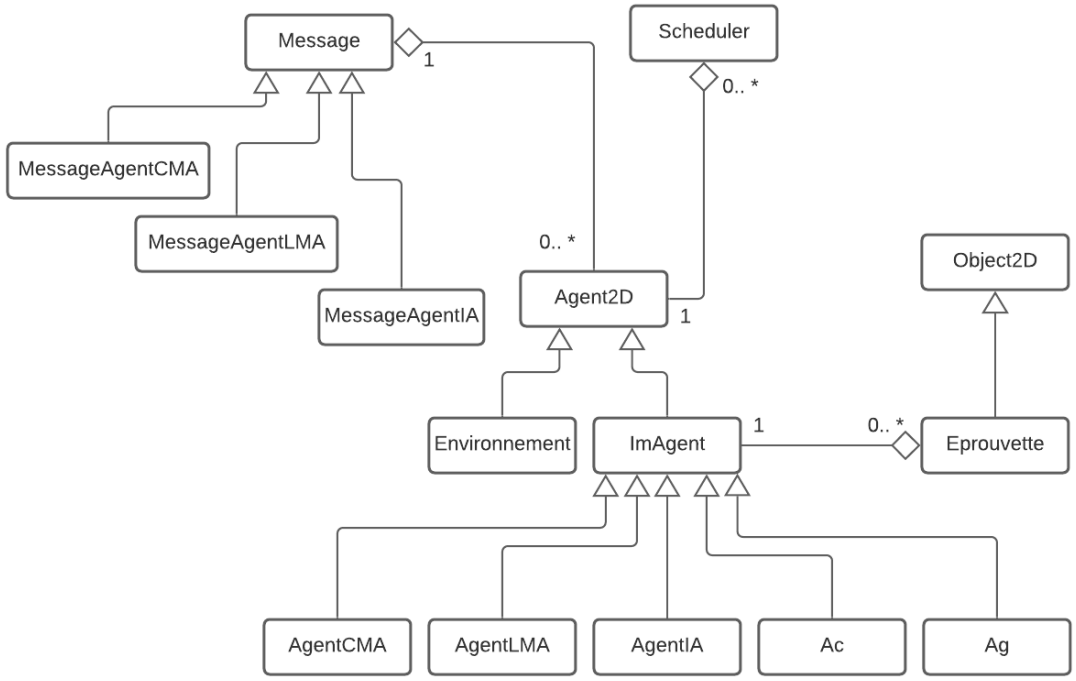

FIGURE 8 – Le diagramme UML de notre projet

```
Un virus de type DoS a ete détecté dans l'agent : AgentIA.1
risk global : 1
```

FIGURE 9 – La simulation commence en avertissant l'utilisateur de la présence d'un virus

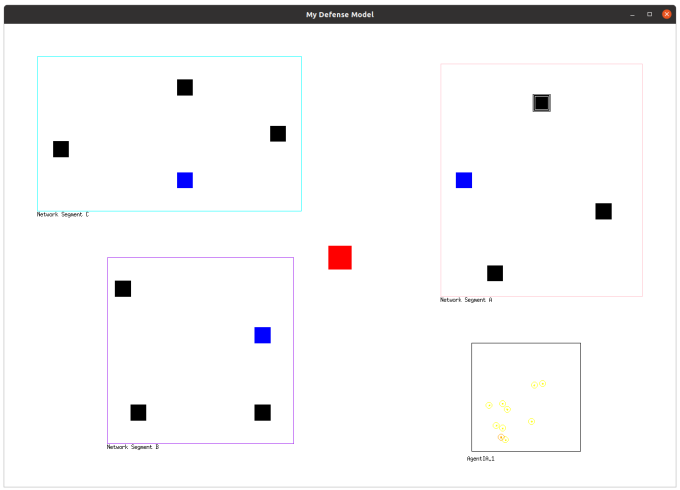

FIGURE 10 – À l'aide de l'éprouvette, une visualisation de l'agent infecté est possible. La menace est rapidement éliminée

```
host changed : AgentIA.2
0
risk global : 0
```

FIGURE 11 – Le risque global est mis à jour une fois la menace écartée

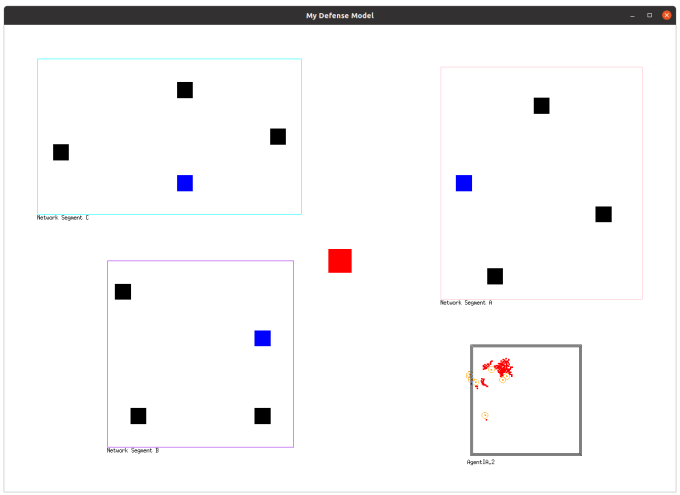

FIGURE 12 – À l'aide des touches 0, 1 et 2, l'utilisateur ajoute des antigènes dans l'agentIA de son choix

```
Creation de Ag.111 de type R2L
Creation de Ag.141 de type brute-force
Creation de Ag.142 de type R2L
Creation de Ag.2330 de type DoS
```

FIGURE 13 – Un message attestant de la création des Ags

```
risk global : 2
```

FIGURE 14 – Le risque global est mis à jour, 2 menaces sont présentes dans le système

8 Améliorations possibles et conclusion

Des améliorations sur ce projet sont possibles au niveau de plusieurs points. Tout d'abord, les agents mémoires sont censés combattre plus efficacement les antigènes qu'ils ont déjà rencontrés, or chaque Ac dans notre simulation a le même comportement, peu importe son type. Nous aurions souhaité modifier la vitesse des Ac mémoires en présence des antigènes qu'ils reconnaissent.

Deuxièmement, le système de vaccin n'a pas été implémenté. Il nous reste à ajouter un message contenant ce vaccin qui serait envoyé de la part des agents LMA aux agents LMA des autres segments.

Enfin, les différents types d'Ag qui peuvent attaquer les agents IA ont tous le même comportement, nous aurions souhaité que chacun présente des caractéristiques différentes (vitesse de propagation, ténacité, etc..) afin de mieux les différencier. Également, une IHM plus travaillée pourrait élargir le type d'utilisateurs qui voudraient utiliser notre projet.

En conclusion, nous pensons avoir bien saisi comment réaliser une simulation d'attaque sur un système réseau peut être améliorer en s'inspirant du fonctionnement de la défense immunitaire du corps humain.

Table des figures

Références

[1] Liu, Nian Liu, Sunjun Li, Rui Liu, Yong. (2009). A Network Intrusion Detection Model Based on Immune Multi-Agent.. IJCNS. 2. 569-574.

[2] Ou, Chung-Ming Ou, C. Wang, Yao-Tien. (2013). Agent-Based Artificial Immune Systems (ABAIS) for Intrusion Detections : Inspiration from Danger Theory. 10.1007/978-3-642-35208-9_4.

[3] Liu, Nian. (2009). A Network Intrusion Detection Model Based on Immune Multi-Agent. Int'l J. of Communications, Network and System Sciences. 02. 569-574. 10.4236/ijcns.2009.26063.