Modélisation de détection d'intrusion par des jeux probabilistes

Mémoire de maîtrise

Présenté par
Madjid Ouharoun

sous la direction de
Prof. Kamel Adi et Prof. Andrzej Pelc

Département d'informatique et d'ingénierie
Université du Québec en Outaouais
283, boulevard Alexandre-Taché
Gatineau (Québec) Canada J9A 1L8

2010

Remerciements

Mes remerciements à ma femme qui m'a fourni un soutient tout au long de mes études.
Je tiens à remercier les professeurs Kamel Adi et Andrzej Pelc pour leur support, leurs directives et leurs conseils durant la réalisation de ce travail.
Je tiens à remercier les professeurs Jurek Czyzowicz et Mohand Said Allili pour avoir accepté d'examiner ce travail.

Table des matières

1 Introduction **5**

2 Revue de la littérature **7**

2.1 Les systèmes de détection d'intrusion 7

2.1.1 Introduction 7

2.1.2 Description 7

2.1.3 Définitions 8

2.1.4 Catégorie d'IDS 10

2.1.5 Sources de données 12

2.1.6 Méthodes d'analyses 13

2.1.7 Fonctionnement des IDS 13

2.2 Évolution des IDS 15

2.2.1 Introduction 15

2.2.2 Exemples 16

2.3 La théorie des jeux 24

2.3.1 Introduction 24

2.3.2 Quelques définitions 24

2.3.3 Représentation 25

2.4 Théorie des jeux et détection d'intrusion 27

3 Présentation du jeu **33**

3.1 Introduction 33

3.2 Présentation du modèle 34

4 Résolution du jeu $J(n, a, 1, g)$ **37**

4.1 Introduction 37

4.2 Formule générale calculant $V(p, q)$ 37

4.2.1 Formule de calcul du gain brut espéré de l'IDS $G(p, q)$ 38

4.2.2 Formule de calcul du coût espéré de l'IDS 38

4.2.3 Calcul du gain net espéré 44

4.3 Résolution du jeu $J(1, 1, 1, g)$ 44

4.4 Résolution du jeu $J(2, 1, 1, g)$ 46

4.4.1 Probabilité de l'activité de l'intrus qui minimise le gain net espéré de l'IDS : $q^*(p)$ 48

4.4.2 Probabilité de l'activité de l'IDS qui maximise la gain net espéré de l'IDS en pire cas : p^* 49

4.5 Résolution du jeu $J(n, 1, 1, g)$ 51

4.6 Version à probabilités dynamiques du jeu $J(n, 1, 1, g)$ 52

4.7 Résolution du jeu $J(2, 2, 1, g)$ 55

4.7.1 Calcul de la formule du gain net espéré $V(p, q) = G(p, q) - C(p, q)$ 55

4.7.2 Probabilité de l'activité de l'intrus qui minimise le gain net espéré de l'IDS : $q^*(p)$ 56

4.7.3 Probabilité de l'activité de l'IDS qui maximise la gain net espéré de l'IDS : p^* 58

5 Résolution du jeu $J(n, n, n, g)$ **61**

5.1 Introduction 61

5.2 Formule générale calculant $V(p, q)$ 61

5.2.1 Calcul du gain brut espéré $G(p, q)$ 61

5.2.2 Calcul du coût espéré de l'IDS $C(p, q)$ 62

5.2.3 Gain net espéré 62

5.3 Résolution du jeu 63

5.3.1 Calcul de $q^*(p)$ 63

5.3.2 Calcul de p^* 63

5.3.3 Conclusion 64

6 Conclusion **65**

Chapitre 1

Introduction

L'avènement des systèmes informatiques, des réseaux et d'Internet a révolutionné la vie quotidienne des individus et des entreprises. La technologie de l'information doit sa popularité à sa facilité d'utilisation. En effet, de nos jours, travailler, naviguer sur le web, consulter son compte bancaire et communiquer via les réseaux et Internet à l'aide d'un ordinateur personnel sont des activités ancrées dans le quotidien. Un monde sans les communications réseau et sans Internet est donc difficilement imaginable.

La popularité de l'outil informatique n'a pas que des avantages. Cela contribue souvent à altérer sa fiabilité. Il est à noter qu'un système informatique fiable doit assurer la disponibilité de ses services, l'intégrité et la confidentialité de ses données. Il est évident que si l'une de ces trois conditions n'était pas satisfaite, on serait confronté soit à un blocage faute de service, soit à de faux résultats, soit à une divulgation de secrets. Dans chacun des trois cas les pertes sont considérables. Pour palier à ce problème les entreprises et les gouvernements investissent considérablement dans la sécurité. La protection des systèmes informatiques est devenue l'une des préoccupations majeures de tous les services informatiques.

Plusieurs solutions sont apparues pour protéger et sécuriser les systèmes informatiques. Elles sont toutes complémentaires mais pas suffisantes. Les antivirus agissent sur une machine hôte et la protègent contre des virus qui sont des programmes capables de s'exécuter pour altérer le bon fonctionnement de la machine. Cette solution est efficace pour des machines isolées et contre des virus déjà connus. D'ailleurs, il est vivement conseillé de maintenir une mise à jour constante des antivirus et de ne pas se fier à eux si on se branche à Internet. En effet, l'antivirus ne détecte pas un virus qu'il ne connaît pas et ne peut rien faire devant les intrusions des systèmes. Pour solutionner le problème de l'intrusion, les pare-feu viennent à la rescousse. Un pare-feu est un

outil matériel ou logiciel utilisé pour contrôler les communications entre un réseau local ou une machine hôte et Internet. Il filtre le trafic dans les deux sens et il bloque les échanges douteux selon une politique de sécurité du réseau. Il est donc l'outil qui définie les logiciels et les personnes qui ont le droit de se connecter à Internet ou à accéder au réseau qu'il protège [14]. Avec l'antivirus, le pare-feu augmente ainsi la sécurisation des données dans un réseau. Mais malgré tout ce qu'on peut penser de leur efficacité combinée, l'antivirus et le pare-feu restent impuissant devant un utilisateur qui répond aux exigences de la politique de sécurité mais qui est animé d'intentions malveillantes. En effet, une fois qu'un logiciel ou un utilisateur a le droit de se connecter à Internet ou à un réseau, rien ne garanti qu'il ne fera pas d'opérations illégales. D'ailleurs des études ont montré que 60% à 70% des attaques proviennent de l'intérieur des systèmes [21]. Pour résoudre ce problème, en plus des antivirus et des pare-feu, les systèmes de détection d'intrusion (IDS) sont utilisés pour surveiller les systèmes informatiques d'une éventuelle intrusion considérée comme étant l'utilisation non autorisée ou l'abus d'utilisation d'un système informatique [26].

Les IDS sont conçus pour retracer les intrusions et protéger les failles du système. Ils sont très efficaces pour reconnaître les intrusions pour les quelles ils sont programmés. Ils sont par contre moins performants si l'intrus change sa façon d'attaquer.

Quelle que soit la performance de l'IDS, elle est souvent limitée par le volume de données que l'IDS peut traiter à la fois. Cette limite ne lui permet pas une surveillance permanente et laisse des brèches aux intrus.

Dans notre recherche, nous proposons de modéliser la problème de la détection d'intrusion comme étant un jeu entre l'intrus et l'IDS selon un modèle probabiliste.

Durant une partie, l'IDS paye un coût unitaire pour chaque activité de vérification d'un paquet réseau. Par ailleurs, il cumule un gain brut lorsqu'il intercepte un paquet malicieux. À la fin de chaque partie, le gain net de l'IDS est calculé par la différence entre le gain brut cumulé et le cumule des coûts unitaires. L'objectif de l'IDS est donc, de trouver une fréquence pour ses activités de vérification qui lui assure le meilleur gain net en pire cas.

Chapitre 2

Revue de la littérature

2.1 Les systèmes de détection d'intrusion

2.1.1 Introduction

Le travail de base sur la détection d'intrusion a été effectué par J.P. Anderson en 1980 [4]. Il a, en effet, révolutionné la sécurité des systèmes informatiques en définissant la problématique du domaine. En 1987, c'est au tour de Denning [11] de mettre au point un modèle de détection, basé sur un système expert, capable de reconnaître certains usages anormaux des systèmes informatiques.
Comme le définissent Rebecca Bace et Peter Mell [5], le système de détection d'intrusion est un outil matériel ou logiciel qui automatise la surveillance des systèmes informatiques dans le but de détecter des signes d'abus dans l'utilisation des ressources des systèmes.

2.1.2 Description

La détection d'intrusion est utilisée essentiellement pour surveiller le fonctionnement des systèmes informatiques. À savoir, les transactions réseaux et l'exécution des applications au niveau des hôtes. La documentation des points faibles du système et la mise sur pied de la politique de sécurité du réseau permettent une programmation adéquate des systèmes de détection d'intrusion. Une fois que l'IDS est installé et configuré, il interagit avec l'environnement dans lequel il est implanté de façon à pouvoir le protéger d'éventuelles intrusions. Durant la phase de détection, il capte l'information via une source de donnée, il l'analyse et il transmet les résultats de l'analyse à l'opérateur qui agit en conséquence, selon qu'il y ait ou pas d'attaques.

Pour décrire les systèmes de détection d'intrusion, nous nous sommes basés sur les travaux du groupe *Intrusion Detection exchange format Workimg Group* (IDWG). Le groupe est l'initiateur de la première tentative pour définir un standard de communication entre les composants d'un système de détection d'intrusion [9].
Partant de ces références, le chantier de la détection d'intrusions s'est diversifié. Dans la suite de cette section nous définissons, dans un premier temps, quelques termes liés à la technologie des IDS. Nous présentons, dans un deuxième temps, les catégories d'IDS. Nous abordons, par la suite, les différentes sources d'information. Cela sera suivi par la présentation des méthodes d'analyses utilisées pour détecter les intrusions. En dernier, nous présentons les modes de fonctionnement des IDS.

2.1.3 Définitions

Les définitions suivantes vont nous permettre de nous familiariser avec certains concepts utilisés dans la technologie des IDS.

Intrusion :

Tous les systèmes de détection d'intrusions partagent une définition générale de l'intrusion comme étant l'utilisation non autorisée ou l'abus d'utilisation d'un système informatique [26].

Système informatique :

Un système informatique est composé d'outils matériels et logiciels qui cohabitent pour le traitement et l'échange de données. Il est composé d'une ou plusieurs machines reliées entre elles à l'aide d'un réseau. Il peut être situé sur un seul site ou sur plusieurs sites éloignés.

Vulnérabilités :

Les vulnérabilités d'un outil informatique représentent tous les bugs de conception ainsi que toutes les lacunes causées par la configuration. Donc les vulnérabilités d'un système informatique représentent la combinaison des vulnérabilités des outils qui le compose. Souvent la présence de certains outils ensemble crée et favorise d'autres vulnérabilités. Toutes ces vulnérabilités représentent des faiblesses pour le système et ainsi des opportunités d'attaques pour les intrus.

Politique de sécurité :

C'est un ensemble de règles établies par les administrateurs des systèmes informatiques. Elles modélisent et formalisent les actions à autoriser et celles à interdire en considérant tous les outils figurant dans le systèmes. Ces autorisations et interdictions sont basées sur les vulnérabilités de chaque outil. Souvent les faiblesses des IDS proviennent du fait qu'ils n'intègrent pas la politique de sécurité dans leurs méthodes d'analyse.

Attaque :

Pour lancer une attaque sur un système informatique, l'intrus collecte dans un premier temps de l'information par le biais d'outils communs comme les scanners de ports. Il exploite ensuite l'information recueillie pour s'introduire dans le système ciblé. Une fois que le système de sécurité est déjoué, l'intrus organise son propre environnement en créant un compte avec tous les privilèges ou en installant des applications de prise de contrôle (cheval de Troie par exemple). Il procède, ensuite, à l'exploration de la cible et enfin il accomplit son méfait.

Signature d'attaque :

Une signature d'attaque est un motif représentant toute l'information concernant une attaque connue. C'est par ce moyen que l'administrateur réseau configure les systèmes de détection d'intrusions. L'exemple suivant représente une signature du système de détection d'intrusion Snort [7].
alert tcp $EXTERNAL_NET any − > $HOME_NET any (msg :"SCAN nmap TCP"; stateless; flags :A,12; ack :0; reference :arachnids,28; classtype :attempted-recon; sid :628; rev :3;)

qui se lit comme suit : si un paquet TCP provenant de l'extérieur ($EXTERNAL_NET) pénètre dans notre réseau ($HOME_NET), peu importe les ports (any), et que ce paquet a le drapeau ACK activé, de même que les deux bits réservés (flags :A,12), et que le numéro d'acquiescement est 0 (ack :0), peu importe l'état de la session (stateless), il faut alors signaler un balayage TCP fait avec nmap.

Alerte :

Une alerte représente l'information transmise par l'IDS à l'intention de l'administrateur. Elle doit être claire, nette et précise.

Faux positif :

On parle de faux positif lorsque l'IDS considère un fonctionnement normal comme une attaque.

Faux négatif :

On parle de faux négatif lorsque l'IDS ne détecte pas une vraie attaque.

LibPcap :

C'est une bibliothèque de fonctions qui sert d'interface à la capture de paquets réseau.

MANET :

C'est un réseau ad hoc mobile, il est constitué de plusieurs nœuds mobiles. Les nœuds communiquent entre eux sans l'aide d'infrastructures fixes. Chaque communication entre deux nœuds se fait durant une session.

2.1.4 Catégorie d'IDS

Il existe deux types de systèmes de détection d'intrusion.

1. Systèmes de détection d'intrusions de type hôte (HIDS) :
   Ces systèmes sont en fait les premiers systèmes mis en œuvre pour la détection d'intrusion. Ils sont installés sur une machine hôte pour la protéger. Thierry Evangelista dit dans son livre « *Les IDS* » ; « *Un HIDS est un agent logiciel que l'on installe généralement sur la machine à protéger et qui analyse en temps réel les flux relatifs à cette machine ainsi que les journaux* »[12]. Ces systèmes sont avantageux lorsque le trafic réseau est chiffré et offre plus de précision dans la surveillance de l'activité sur l'hôte. Par contre, ils sont moins performants contre les attaques de déni de service et les scans. Le déni de service se produit quand le serveur est submergé par des requêtes et qu'il n'arrive pas à répondre. Les scans sont de simples requêtes qui permettent à l'attaquant de savoir quels sont les ports ouverts sur une machine et ainsi déduire les services disponibles.

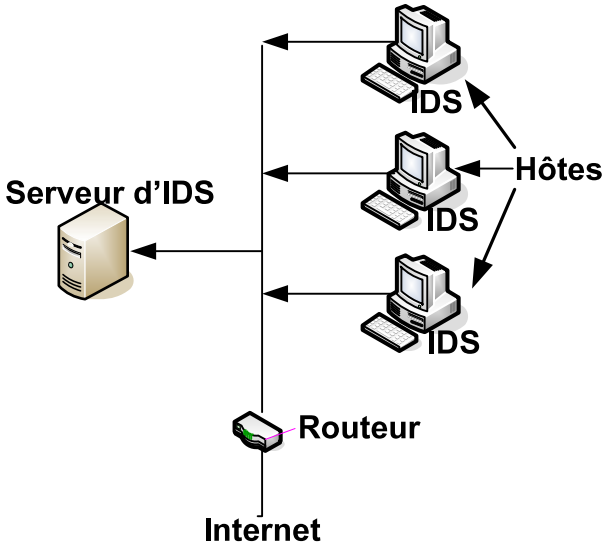

FIGURE 2.1 – Exemple de HIDS

2. Systèmes de détection des intrusions réseaux (NIDS) :

   Ces systèmes sont mis en œuvre pour la protection des réseaux. Ils se basent sur le principe de l'analyse du trafic réseau. Ils sont composés de sondes (capteurs) qui surveillent les données acheminées dans le réseau et d'un moteur pour analyser les données [15]. L'architecture du réseau et la politique de sécurité permettent de définir l'emplacement des sondes. Les NIDS sont efficaces contre les scans, mais ils sont confrontés au problème des réseaux cryptés.

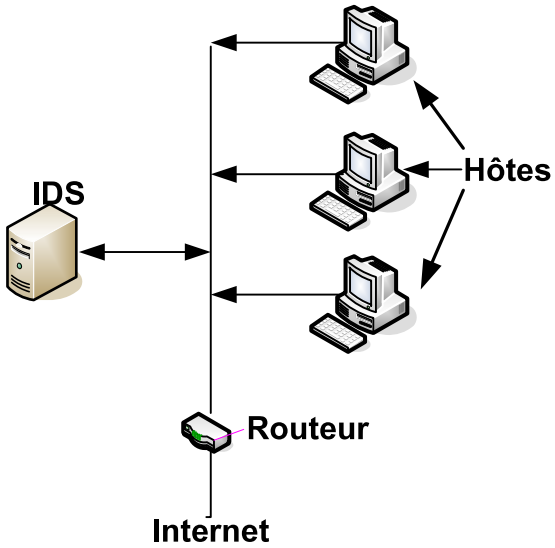

FIGURE 2.2 – Exemple de NIDS

2.1.5 Sources de données

Pour surveiller l'infrastructure informatique les systèmes de détection d'intrusions comptent sur une ou plusieurs sources de données qui fournissent l'information à analyser. Une source de données fournie un rapport sur les activités qui se sont produites dans le système à surveiller. Dans[10] Debar l'a définie comme étant un flux de données acquis par le système de détection d'intrusion et analysé pour déterminer si des événements anormaux par rapport à l'algorithme d'analyse se produisent dans ces données. Cependant ce flux de données peut provenir de plusieurs points d'un système informatique et la détection d'intrusion considère trois sources importantes :

1. Le trafic réseau :

   Le trafic réseau constitue la source de données pour les sondes NIDS. Ces sondes placées sur le réseau en mode promiscuité captent et analysent les données passant à travers le réseau pour détecter d'éventuelles attaques. Les avantages quant à l'utilisation de cette source de données sont multiples. La promiscuité de la sonde constitue un atout majeur, elle empêche sa détection et annule son impact sur le système surveillé. Une sonde NIDS surveille une portion d'un réseau, elle offre donc la possibilité de détecter des attaques combinées. Les données utilisées sont dans un format standard et donc plus accessibles. Il est aussi à noter que les sondes NIDS sont simples à installer et à utiliser. Toutefois, cette technique basée sur les échanges réseaux se retrouve impuissante devant des attaques qui n'impliquent pas un trafic réseau. Il est aussi à signaler que le débit des réseaux, de nos jours, empêche le traitement exhaustif des données [21].

2. Les données systèmes :

   Elles représentent les fichiers d'audits produits par les systèmes d'exploitations où sont installés les sondes HIDS [21]. La force de cette technique réside dans la fiabilité et la richesse des données. En effet, les fichiers d'audits reflètent ce qui s'est réellement passé sur une machine. La faiblesse par contre de cette technique réside quant à elle dans :

   (a) Le volume important des données à traiter.

   (b) Le fait d'être limitée à la surveillance d'une seule machine.

3. L'audit applicatifs :

   Au lieu de traiter toutes les interactions qui se produisent sur une machine, l'audit applicatif favorise le ciblage de certaines applications (serveur web, ftp etc). Cette technique présente l'avantage de traiter des données spécifiques avec un volume réduit. Toutefois, les applications en question doivent être configurées pour produire des fichiers d'audit [21].

2.1.6 Méthodes d'analyses

La technologie des IDS permet d'analyser les données recueillies de trois façons :

1. Analyse centralisée : L'IDS possède plusieurs capteurs, il centralise les alertes pour les analyser sur une seule machine. Ce type d'analyse présente l'avantage d'avoir une vue globale sur toutes les machines protégées. Toutefois, elle à l'inconvénient de l'occupation très longue du réseau pour acheminer l'information.
2. Analyse locale : Chaque machine dispose d'un capteur et analyse l'information à son niveau. Avec ce type d'analyse le trafic réseau est diminué mais les attaques distribuées peuvent échapper à la détection.
3. Analyse distribuée : Des petits programmes appelés « *agents* » sont déployés sur les nœuds du réseau. Pour les besoins d'analyse un agent est envoyé sur une machine pour traiter l'information.

2.1.7 Fonctionnement des IDS

Les IDS fonctionnent suivant deux approches.

1. Approche par scénario (anomaly detection) :
   Elle ressemble beaucoup aux techniques utilisées par les antivirus. Elle est basée sur la notion de signatures d'attaques. Comme on l'a vu dans la section 2.1.3, la signature d'une attaque représente les caractéristiques de cette attaque. La technique consiste donc, à analyser les flux de données en les comparant aux signatures pour identifier d'éventuelles intrusions.
   Il existe plusieurs mécanismes pour mettre en oeuvre cette approche.
   Voici trois d'entre eux :
   (a) Analyse par comparaison (Pattern Matching) :
   Le principe de cette approche est de faire correspondre à chaque signature d'attaque un motif (Pattern) qui est sous forme d'une chaîne de caractères. Durant l'analyse du flux de données qui est aussi une chaîne de caractères, le système de détection d'intrusion tente de reconnaître les motifs d'attaques déjà connus [20].
   (b) Système expert :
   Technique qui repose sur une base de connaissances et un moteur d'inférence. La base de connaissances est composée elle-même d'une base de règles décrivant les attaques et d'une base de faits contenant les événements relatifs aux attaques. Durant la phase

de détection, le moteur d'inférence est capable de détecter les attaques en utilisant la base des connaissances [20].

(c) Analyse de transition d'états :

Dans cette approche on représente les attaques sous forme d'un ensemble d'états par lesquels passe le système à surveiller. Les états sont définis par des conditions sur les variables du système. Les transactions représentent les actions suivant les événements qui surviennent.

Le plus grand inconvénient de l'approche par scénario réside dans le fait qu'elle ne détecte que les attaques connues. Elle est impuissante devant de nouvelles attaques [13]

2. Approche comportementale (misuse detection) :

Contrairement à l'approche par scénario cette approche se base sur le comportement passé des entités comme les utilisateurs, les applications et les services. Le principe repose sur la modélisation de ces entités pour mieux les contrôler. En effet, dans un premier temps, on fait correspondre un profil à chaque entité en se basant sur son comportement normal. Dans un deuxième temps, pendant la phase de détection, on observe l'entité modélisée et tous les événements qui ne sont pas représentés dans le profil, déclenchent des alertes d'attaques [24].

Pour faire correspondre un profil à chaque entité on a besoin de la politique de sécurité et d'une phase d'apprentissage. Initialement les profils ne correspondent qu'à la politique de sécurité. Durant la phase d'apprentissage, les profils sont améliorés en définissant le comportement normal de chaque entité. Cette phase d'apprentissage peut être limitée dans le temps ou bien continue tout au long de l'exploitation. La mise en œuvre effective de la détection d'anomalies dépend beaucoup de l'approche utilisée pour construire les profils. Nous présentons ici les trois approches les plus populaires.

(a) Approche probabiliste :

Dans cette approche la construction des profils se base sur la probabilité qu'un événement ait lieu par rapport à une séquence d'autres événements.

(b) Approche statistique :

Ici la construction des profils se base sur des mesures quantitatives de l'utilisation des ressources systèmes.

(c) Approche par réseau de neurones :

Cette approche se base sur le comportement de chaque utilisateur. Le profil normal d'un utilisateur est construit en prenant en compte les activités de l'utilisateur comme

les outils préférés, les habitudes de travail, la vitesse de frappe au clavier, etc. Le profil est ensuite représenté par un réseau de neurones qui enregistre les opérations de l'utilisateur durant une fenêtre temporelle et il tente de prédire la prochaine opération [20].

En utilisant l'approche comportementale, la détection d'intrusions puise sa force dans l'habilité à détecter des attaques inconnues.

Quelle que soit l'approche utilisée, les IDS peuvent déclencher une alerte en l'absence d'attaque (faux positif) ou encore pas d'alerte en présence d'attaque (faux négatif). Ainsi entre les deux il n'est pas facile de détecter la vraie intrusion.

2.2 Évolution des IDS

2.2.1 Introduction

Comme on l'a mentionné plus haut, il existe deux approches d'analyse pour les IDS, l'approche comportementale et l'approche par scénarios. La première approche qui est aussi la plus ancienne cherche à détecter les comportements anormaux dans le système par rapport à un comportement normal, appelé communément *profil normal*, défini et modélisé au préalable [25]. La définition et la modélisation de ce dernier nécessitent une phase d'apprentissage pour assoire une ligne de conduite qui devrait être respectée par le système sous la surveillance de l'IDS. Une fois que le profil normal est modélisé, l'IDS rentre dans sa phase de détection et une alerte est donnée à chaque fois que le système dévie de ce profil. Toutefois l'IDS tolère un certain taux de déviation selon l'approche utilisée pour déterminer le profil normal.

L'avantage majeur de cette technique est qu'elle est très efficace pour détecter les attaques inconnues. Elle limite beaucoup les faux négatifs. Cependant elle peut produire beaucoup de faux positifs en cas d'évolution du système sans reprendre la phase d'apprentissage. Un autre inconvénient peut apparaître dans le cas d'un utilisateur malveillant qui modifie progressivement son profil, sans dépasser le seuil de tolérance. D'ailleurs, il est souvent très ardu de débusquer ce genre d'intrus.

L'approche par scénarios, par contre, adopte une autre philosophie. Elle ne se préoccupe pas du comportement du système mais plutôt des techniques des attaquants. En effet, un IDS fonctionnant avec cette approche, utilise une base de signatures représentant les scénarios d'attaques

connues et identifiées au préalable. La détection effective se réalise en comparant les informations fournies par la source de données avec les scénarios de la base des signatures. Toute concordance entre les deux déclenche une alerte. Le point fort de cette approche réside dans la réduction des faux positifs. Du côté des points faibles, on peut citer son impuissance devant les nouvelles attaques et le format générique des signatures. En effet, une attaque n'est pas toujours réalisée de la même façon et une nouvelle attaque est toujours inconnue donc non répertoriée dans la base des signatures.

Quelle que soit la technique d'analyse utilisée, la tâche de l'IDS et de signaler à l'administrateur les activités supposées anormales. L'administrateur par la suite analyse cette information et prend les mesures adéquates. Cette analyse qui n'est toujours pas automatisée reste la tâche la plus importante et la plus contraignante dans la détection d'intrusion. Elle est importante car elle permet de prendre des mesures pour protéger l'infrastructure surveillée. Elle est aussi contraignante pour l'administrateur qui doit analyser un volume très important d'informations souvent génériques et différentes selon la nature de l'intrusion, du type d'IDS et de l'emplacement des sondes. Au fil du temps on s'est rendu compte que plusieurs facteurs rentrent en ligne de mire dans l'efficacité de la détection d'intrusions. Il est bien beau de fortifier un système informatique par des IDS qui nous informent du moindre détail qui s'y produit, mais la réalité est toute autre. Les alertes sont souvent nombreuses, redondantes, parfois de même nature mais différentes et surtout génériques. Face à toutes ces contraintes, l'administrateur se retrouve toujours devant un casse tête pour débusquer les intrus. Plusieurs recherches ont étudié le problème sous différentes approches. Dans la suite de ce document nous présenterons quelques-unes des ces approches.

2.2.2 Exemples

Snort

Snort est un système de détection d'intrusions à temps réel très efficace pour détecter les attaques se déroulant sur un seul paquet. Comme le montre la figure 2.3, Snort utilise la librairie LibPcap pour capturer les paquets réseaux. Le paquet ainsi récupéré sera traité dans un premier temps par un module de décodage pour détecter les différents protocoles. Une fois le décodage terminé, les préprocesseurs prennent le relais. À l'origine, ces derniers s'occupaient uniquement du formatage de données pour les rendre compatibles avec la base des règles. Mais pour plus d'efficacité, des programmeurs ont ajouté des fonctionnalités supplémentaires (balayage des ports, gestion des sessions, etc.). C'est d'ailleurs pour cette raison que les préprocesseurs communiquent

directement avec les modules d'affichage. L'engin de détection, composé d'un module de détection et d'une base de signatures, traite l'information transmise par les préprocesseurs pour détecter d'éventuelles attaques. Et enfin les modules d'affichage servent à interagir avec l'utilisateur.

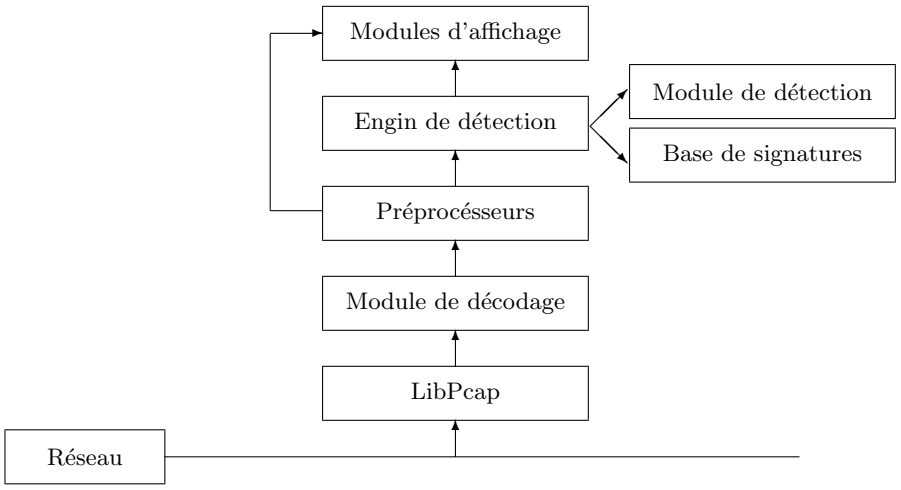

FIGURE 2.3 – Les différents modules de Snort à l'origine

Détection d'attaques se déroulant sur plusieurs paquets

Partant de Snort Mathieu Couture [7] a amélioré la syntaxe du langage de signatures pour représenter les attaques se déroulant sur plusieurs paquets. Comme illustré à la figure 2.4, il a rajouté un nouveau système de détection de scénarios qui utilise les événements fournis par l'engin de détection de Snort. Les résultats de ce nouveau traitement servent à mettre à jour une base de connaissances qui représente le contexte des attaques. Les attaques sont représentées à l'aide d'un langage qui permet de décrire les paquets les constituant ainsi que les liens entre ces paquets. Ce langage permet aussi l'interaction avec la base des connaissances pour la mettre à jour. Cette dernière contiendra les renseignements sur :

- les sessions TCP actives,
- les systèmes d'exploitation existants,
- les services offerts par chaque poste,
- la politique de sécurité.

Dans cette nouvelle optique, les préprocesseurs se chargent uniquement de la régularisation des paquets. Le nouveau module de détection utilisera les résultats du module de détection de Snort, la base de scénarios et la base de connaissances pour détecter les attaques et éventuellement

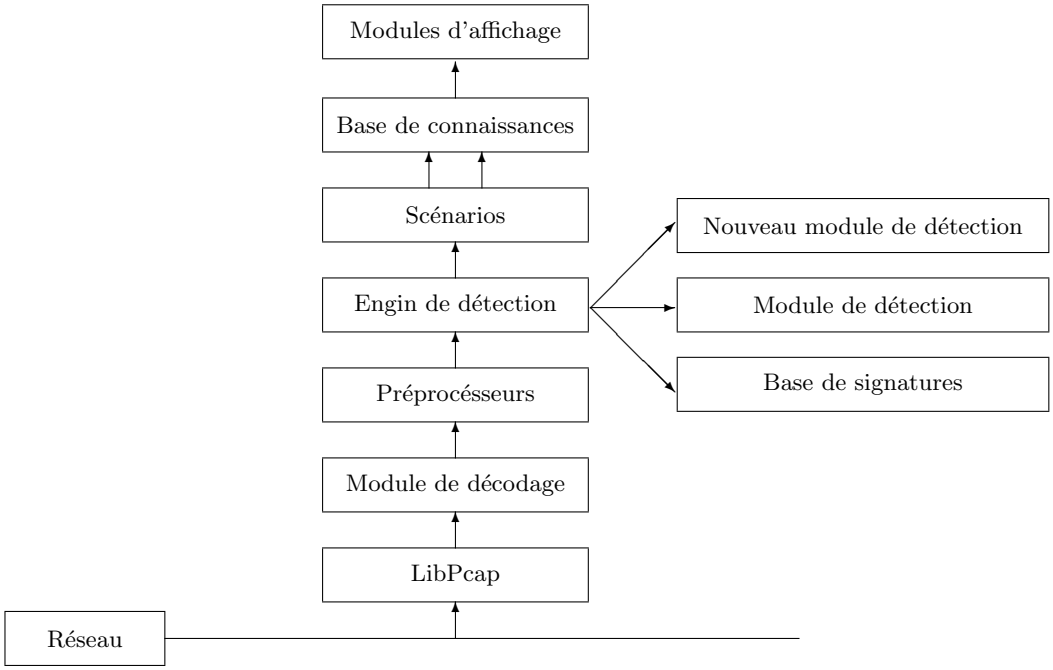

FIGURE 2.4 – Les modules de Snort après l'apport de M. Couture.

mettre à jour la base de connaissances. Le modèle utilisé se base essentiellement sur les définitions suivantes :

1. Un entête $\eta = \langle \nu_1, \nu_2, ... \rangle$ est un tuple de valeurs, dont la première représente le nom du protocole auquel elle est associée. Le tuple suivant $\langle ip, 132.203.250.26, 142.92.39.88, 246, ... \rangle$ représente un entête IP, spécifié par le premier élément du tuple, dont l'adresse source est 132.203.250.26, la destination 142.92.39.88, le *time to live* a une valeur de 246, etc.
2. Un paquet $\rho = \langle \{\eta_1, \eta_2, ...\}, \tau \rangle$, est un ensemble d'entêtes, auxquelles on adjoint un temps τ.
3. Une trace $\sigma = \rho_0 \rho_1 ...$ est une suite infinie de paquets.
4. Par σ(i), on désigne le paquet ρ_i, et par σ^i, on désigne le suffixe de σ commençant à ρ_i.

Le modèle utilisé par Mathieu Couture [7] s'inspire de la syntaxe suivante :

$\Psi ::= \text{tt} \mid \langle id, \Phi \rangle \Psi \mid \neg\Psi \mid \Psi_1 \wedge \Psi_2$ (Formule pour les traces)

$\Phi ::= \text{p} \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2$ (Formule pour les paquets)

La première partie exprime les scénarios :

– tt formule logique vraie pour toute trace, elle représente le scénario où il n'y a rien de spécial qui se passe.

– $\langle id, \Phi \rangle\ \Psi$ représente le scénario débutant par le paquet id vérifiant les caractéristiques Φ et qui se poursuit par le scénario Ψ.
– $\neg\Psi$ veut dire que le scénario Ψ ne survient pas.
– $\Psi_1 \wedge \Psi_2$ représente le scénario où Ψ_1 et Ψ_2 se produisent tous les deux.

La deuxième partie représente les paquets et elle est exprimée par calcul propositionnel.
À partir de ce modèle découle la syntaxe du langage de signatures utilisé.

$\langle specification \rangle$: := $\langle scenarios \rangle$

$\langle scenarios \rangle$: := $\langle step \rangle (\langle step \rangle \langle timeout \rangle)^*$

$\langle step \rangle$: := $\langle header \rangle \langle filter \rangle \langle output \rangle^*$

$\langle header \rangle$: := step$\langle id \rangle$: $\langle snortsig \rangle$

$\langle filter \rangle$: := match : $\langle matchcrule \rangle^*$

$\langle matchcrule \rangle$: := $\langle id \rangle.\langle field \rangle \langle op \rangle\ \langle id \rangle.\langle fieled \rangle$;|$\langle prolog \rangle$;

$\langle op \rangle$: := $< \mid \leq \mid > \mid \geq \mid =$

$\langle output \rangle$: := output : $\langle prolog \rangle$

$\langle timeout \rangle$: := timeout : $\langle id \rangle$.timestamp + $\langle time \rangle\ \langle output \rangle^*$

$\langle time \rangle$: := $\langle number \rangle \langle timeunit \rangle$

$\langle timeunit \rangle$: := sec | hours | days | weeks | years |

On remarque qu'avec cette syntaxe, une attaque peut être exprimée par un ou plusieurs scénarios. Un scénario est spécifié par une ou plusieurs étapes qui sont en fait des paquets réseaux. Si un scénario est représenté par plusieurs paquets, un intervalle de temps entre eux doit être respecté $\langle timeout \rangle$. Un paquet est constitué par son identifiant $\langle id \rangle$ est une signature Snort. Un filtre est utilisé pour lier les paquets entre eux et vérifier les conditions d'une attaque ou de tout autre événement. Le nom terminal $\langle output \rangle$ sert à afficher les alertes et mettre à jour la base de connaissances.

Pour illustrer les résultats mis en avant nous donneront ici deux exemples de scénarios. Le premier détecte les sessions actives et active l'alerte «synscan». Le second détecte l'adresse du routeur.

Exemple1 : Ports TCP ouverts et sessions actives [7] :

1. step syn : tcp (flags :S ;)
2. step synack : tcp (flags :SA ;)
3. match : syn.sip=synack.dip ; syn.dip=synack.sip ; syn.sport=synack.dport ; syn.dport=synack.sport ;

4. output : assert(port(syn.dip,syn.dport,open)).
5. timeout : syn.timestamp+2sec ;
6. step synack : ack (flags :A ;)
7. match : syn.sip=ack.sip ; syn.dip=ack.dip ; syn.sport=ack.sport ; syn.dport=ack.dport ;
8. output : assert(session(syn.sip,syn.dip,syn.sport,syn.dport)),
9. assert(session(syn.dip,syn.sip,syn.dport,syn.sport)).
10. timeout : synack.timestamp+2sec ;
11. output : assert(alert("synscan",syn.sip,syn.dip)).

Explication :
Ligne1 : un utilisateur A demande une connexion à l'utilisateur B
Ligne2 et 3 : l'utilisateur B accepte la connexion
Ligne4 : affirmation que le port de communication de B est ouvert
Ligne5, 6, 7, 8, 9, 10 et 11 : si dans un intervalle de 2 secondes après que A a lancé sa première demande, il confirme la connexion alors il faut affirmer que la session est ouverte. Mais dans le cas contraire il y a affirmation du ≪scan≫ des ports.

Exemple2 : Adresse du routeur [7] :

1. step anyip1 : ip
2. step anyip2 : ip
3. match : anyip1.smac=anyip2.smac ; anyip1.sip !=anyip2.sip ;
4. timeout : anyip1.timestamp + 10sec ;
5. step anyip3 : ip
6. match : anyip1.smac=anyip3.smac ; anyip1.sip !=anyip3.sip ; anyip2.sip !=anyip3.sip ;
7. output : assert(gatewaymac(anyip1.smac)).
8. timeout : anyip2.timestamp + 10sec ;

Explication : Si plusieurs paquets passant dans le réseau ont la même adresse MAC mais des adresses IP différentes, on peut conclure que cette adresse MAC représente un routeur.

Détection des variations d'attaques

Pierre-Luc Lesperance [16] part du principe que les NIDS basés sur l'approche par scénarios nécessitent une signature propre à chaque attaque, et qu'une variante aussi minime qu'elle soit ne sera jamais détectée.

Exemple[16] : Soit une signature vérifiant que l'utilisateur root ne peut pas utiliser le service FTP. L'IDS en surveillant le port 21 attend la chaîne "USER root". L'intrus par contre envoie les trois paquets suivants :

P1 = "USER roo"

P2 = "o"

P3 = "t"

La figure 2.5 illustre ce qui se déroule au niveau de l'attaquant, de la cible et de l'IDS. L'attaque ne sera pas détectée puisque l'IDS n'a aucune connaissance des commandes système. Donc lorsqu'il assemble les trois paquets il aura la commande "USER rooot" qui ne correspond à aucune signature. Par contre le destinataire qui a la possibilité de corriger les commandes, obtiendra "USER root"

Pour résoudre ce problème l'auteur a proposé un module de comparaison se basant sur le calcul de distance entre le flux de données et la signature. À chaque message reconstitué par l'IDS, le nouveau module le compare avec toutes les signatures. S'il retrouve une signature dont la ressemblance est établie selon un seuil de tolérance, une alerte sera déclenchée. Le plus grand défi de cette solution réside dans le choix du seuil de tolérance.

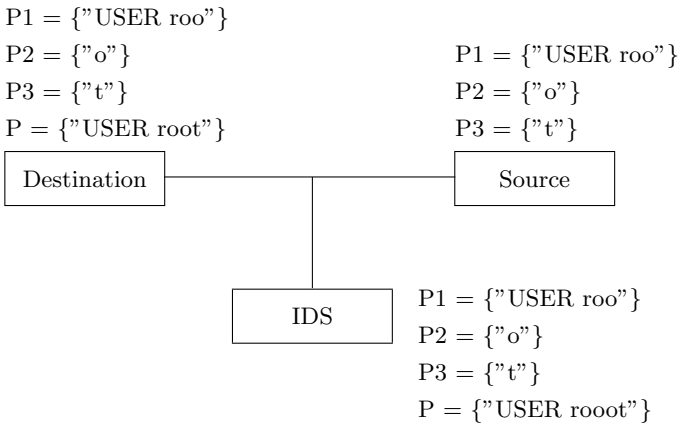

FIGURE 2.5 – Attaque sur plusieurs paquets

Détection d'anomalies basée sur la politique de sécurité

Dans [25], Jakub Zimmermann propose une méthode de détection d'anomalies basée uniquement sur la politique de sécurité sans modélisation du comportement de l'utilisateur et des applications. Cette méthode ne passe pas par une phase d'apprentissage mais s'inspire uniquement d'une définition formelle de la politique de sécurité. Il a en effet modélisé le système sous forme d'objets et de flux d'informations possibles entre ces objets. Chaque flux d'informations représente une opération exécutée entre deux objets du système. Tous les flux légaux forgent donc la politique de sécurité. Dans les faits, pour modéliser cette politique de sécurité il s'est inspiré du système de contrôle d'accès.

Durant la phase d'exploitation, le système de détection d'intrusion surveille les opérations entre les différents objets et celle qui engendre un flux illégal viole la politique et déclenche une alerte. Les tests réalisés sur cette solution ont donné des résultats satisfaisants pour une machine isolée mais elle reste impuissante devant les attaques réseaux.

Corrélation d'alertes

Les systèmes utilisés actuellement dans la détection d'intrusion produisent beaucoup d'alertes qui sont inutiles. On retrouve surtout beaucoup d'alertes redondantes et de faux positifs. Plusieurs recherches ont été faites pour résoudre ce problème. Nous présenterons dans ce qui suit la solution proposée par Frédéric Cuppens[8].

Son travail se base sur l'utilisation dans un système informatique de plusieurs types d'IDS qui alertent le gestionnaire en temps réel sur les attaques identifiées. Chaque IDS surveille une portion du réseau, ce qui augmente les chances de détection d'attaques contre le système. En plus des fonctions propres aux différents IDS, l'auteur a développé cinq fonctions pour assurer la coopération entre les IDS. La figure 2.6 présente l'architecture de la solution proposée.

- Fonction de gestion des alertes (Alert base management function) :
  Cette fonction reçoit en entrée les alertes des différents IDS sous le format IDMEF(Intrusion Detection Message Exchange Format), les converti en un ensemble de tuples et les sauvegarde dans une base de données relationnelle.
- Regroupement d'alertes(Alert Clustring) :
  La fonction de regroupement prend comme entrée les éléments de la base de données créée par la fonction précédente. Elle crée ensuite des groupes d'alertes de sorte que les alertes d'un même groupe correspondent à la même attaque. Pour ce faire l'auteur a utilisé un système expert pour le calcul de similarité entre les alertes.

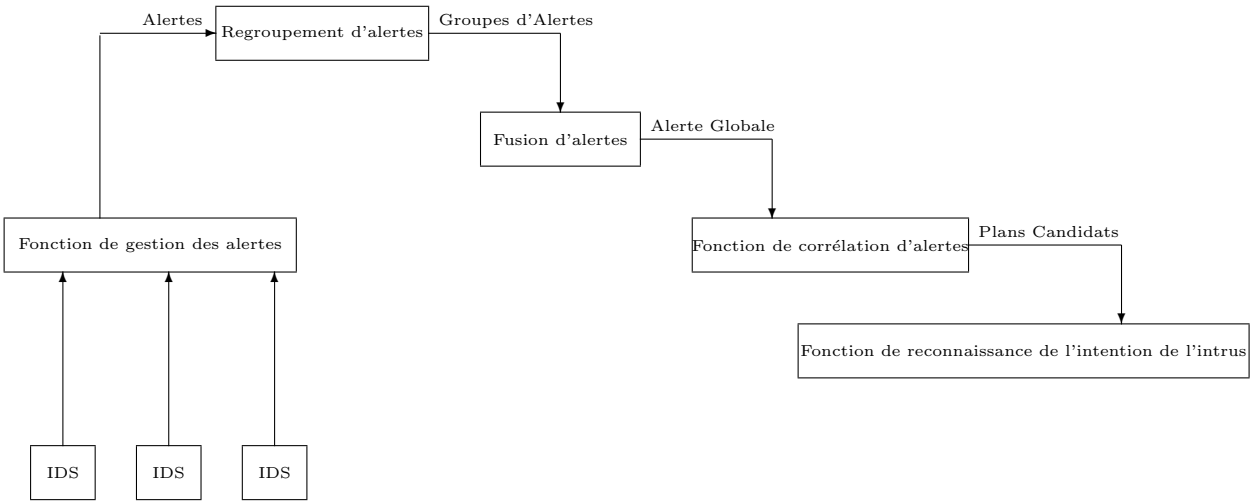

FIGURE 2.6 – Schéma général du module de coopération

- Fusion d'alertes(Alert Merging) : Le rôle principal de cette fonction comme son nom l'indique est de fusionner les alertes de chaque groupe en une alerte représentative du groupe. En fait cette fonction reçoit, de la fonction de classification, des groupes d'alertes. Pour identifier les alertes appartenant à un groupe, l'auteur à introduit le prédicat *cluste_alert(clusterid,alerteid)* qui signifie que l'alerte identifiée par *alerteid* appartienait au groupe identifié par *clusterid*. La fonction de fusion génère pour chaque groupe une alerte globale représentée par le prédicat *cluster_global_alert(clusterid,alerteid)* qui signifie que l'alerte identifiée par *alerteid* est l'alerte globale du groupe identifié par *clusterid*. L'alerte globale est générée en rassemblant le plus d'information présente dans les alertes.
- Fonction de corrélation d'alertes (Alert Correlation) :
  Cette fonction assure la corrélation des alertes produites par la fusion dans le but de déjouer les scénarios d'attaques. Elle se base sur le principe qu'un intrus, pour atteindre son objectif, met en œuvre plusieurs attaques. La détection d'intrusions classique ne permet de détecter que les attaques élémentaires. Cette corrélation a donc pour objectif de retracer les plans d'intrusions mis sur pied par l'intrus. Ces derniers sont appelés *Plans Candidats*.
- Fonction de reconnaissance de l'intention de l'intrus (Intention Recognition) :
  Deux cas peuvent se présenter avec ces *Plans Candidats*. Le cas où on reconnaît que l'intrus a atteint son objectif, l'intrusion sera signalée au gestionnaire. Mais on peut être dans le cas où les plans candidats indiquent que l'objectif n'est pas atteint. À ce moment là, la fonction de reconnaissance de l'intention fait une extrapolation des plans candidats pour anticiper l'intention de l'intrus. Elle fait une analyse des actions passées de l'intrus, des résultats obtenus et suggère la suite qu'il peut donner.

2.3 La théorie des jeux

2.3.1 Introduction

La théorie des jeux est une théorie mathématique qui permet de modéliser des situations où la prise de décision est interactive. Deux ou plusieurs intervenants (joueurs), à intérêts divergents, prennent des décisions, agissent et participent à l'issue de chaque partie. Chaque joueur intervient pour ramener la partie en sa faveur. Les joueurs sont considérés comme étant rationnels et chacun agît en prenant en compte les actions possibles des autres. C'est au début du XXe siècle que sont apparus les premiers travaux sur les jeux de stratégies avec Zermelo (1912), Borel (1921), Von Neumann (1928). Mais la théorie des jeux est concrètement née en 1944 avec l'ouvrage : ≪*Theory of Games and Economic Behavior*≫ de Von Neumann et Morgenstern. Depuis les travaux de John Nash, vers les années 1950, qui ont donné la notion de solution aux jeux à somme non-nulle, la théorie des jeux a connu un développement intéressant. On a vu de nombreuses applications en biologie, en économie, en informatique, etc[6].
Dans [23], Daniel Schneider présente la théorie des jeux comme étant un outil qui sert à modéliser des situations où des acteurs sociaux prennent des décisions individuelles séparées, mais ayant un impact combiné sur les acteurs. Cela veut dire que l'issue d'une partie dépend des démarches suivies par l'ensemble des joueurs.

2.3.2 Quelques définitions

Stratégie

En théorie des jeux, une stratégie est définie comme étant l'action d'un joueur. Généralement on parle de deux sortes de stratégies. D'un côté on a les **stratégies pures** qui sont des actions ou des plans d'actions choisies avec certitude par chaque joueur. De l'autre côté, on fait intervenir un mécanisme aléatoire qui affecte un poids à chaque stratégie pure pour obtenir des **stratégies mixtes**.

Équilibre de Nash

On dit qu'un jeu à n joueurs possède un équilibre de Nash s'il existe un n-uplet de stratégies tel qu'aucun joueur n'a intérêt à changer unilatéralement sa stratégie. En d'autres termes, soient :

- $N = \{1, 2, .., n\}$: ensembles des joueurs.

- s_i : stratégie pure du joueur i.
- S_i : ensemble des stratégies du joueur i.
- $s = (s_1,s_2,...s_n)$: combinaison de stratégies à raison d'une par joueur.
- $s_{-i} \in S_{-i}$: toutes les stratégies sauf celle de i.
- $u_i \in R$: fonction de gain du joueur i.

On dit que la combinaison de stratégies $s^* = (s_1^*,s_2^*,...s_n^*)$ représente un équilibre de Nash si $\forall i \in N, s_i \in S_i, u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$

Jeux coopératifs et non coopératifs

Dans un jeu coopératif, les joueurs peuvent communiquer entre eux et passer des accords. L'issue respecte l'intérêt général avec le partage des gains entre les joueurs. Dans un jeu non coopératif les joueurs ne communiquent pas. Chacun agît dans son propre intérêt pour faire balancer le jeu à son avantage.

Jeux à somme nulle et non nulle

On dit qu'un jeu est à somme nulle lorsque le gain d'un joueur représente exactement la perte d'un autre. Mais avec ce type de jeux ce n'est pas toujours évident de représenter la réalité. C'est pour cette raison qu'on utilise les jeux à somme non nulle pour modéliser beaucoup de cas réels.

2.3.3 Représentation

Un jeu est régie par des règles et fait intervenir au moins deux joueurs. Ces derniers se disputent l'issue de chaque partie en suivant des stratégies. Durant chaque partie, une fonction de paiement octroie des gains à chaque participant. La finalité de chaque joueur est d'avoir le gain le plus élevé à l'issue de chaque partie.
La théorie des jeux offre deux façons pour représenter un jeu, la forme normale et la forme extensive. La forme normale offre une représentation étape par étape des stratégies et de la fonction de paiement de chaque joueur. Toutefois, la forme normale n'est possible que pour des jeux simultanés où le nombre de joueurs ainsi que l'ensemble des stratégies sont finis. Quant à la forme extensive, elle permet une représentation sous forme d'arbre. Les nœuds symbolisent les positions du jeu, les transitions entre les nœuds représentent les actions des joueurs et les feuilles donnent les gains.

Exemple : Le dilemme du prisonnier

Le dilemme du prisonnier est un exemple célèbre de la théorie des jeux. Deux prisonniers complices d'un délit sont retenus dans deux cellules séparées et ils ne peuvent pas communiquer.

- Si un prisonnier dénonce son complice alors que ce dernier ne le dénonce pas, le premier est remis en liberté, tandis que le second obtient une peine de 10 ans.
- Si les deux prisonniers se dénoncent mutuellement, ils sont condamnés chacun à 5 ans de prison.
- Si les deux refusent de se dénoncer mutuellement, ils ont une peine de 3 ans chacun, faute de preuves.

Prisonniers	dénonce	se tait
dénonce	(-5,-5)	(0,-10)
se tait	(-10,0)	(-3,-3)

FIGURE 2.7 – Représentation sous la forme normale du dilemme du prisonnier

La figure 2.8 suivante donne exactement la même information que le tableau précédent mais sous forme d'arbre.

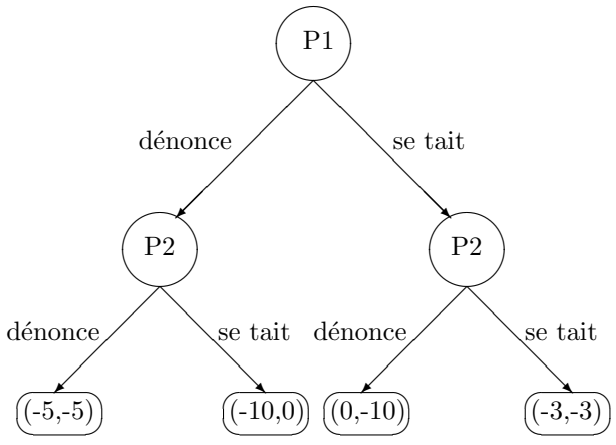

FIGURE 2.8 – Représentation sous la forme extensive du dilemme du prisonnier

2.4 Théorie des jeux et détection d'intrusion

La théorie des jeux appliquée au problème de la détection d'intrusion commence à faire son chemin depuis quelques années. Plusieurs travaux de recherche [1,2,3,17,18,19 et 21] se sont intéressés au problème. Dans cette section, nous allons présenter quelques-unes de ces contributions.
Dans [1] les auteurs définissent un jeu entre l'intrus et l'IDS. L'intrus est soit un attaquant utilisant un ou plusieurs noeuds du réseau, soit une coalition d'attaquants animés par un même objectif. Même si les auteurs introduisent la notion de coalition d'attaquants, cette dernière n'est pas exploitée pour construire des attaques. En effet, l'attaque est sur un seul paquet, donc émise d'un seul nœud. Pour modéliser le jeu, les auteurs ont considéré le système comme suit :

- Le réseau est un ensemble $T = \{t_1, t_2, ..., t_M\}$ de sous-systèmes, sachant qu'un sous système peut être une portion du réseau, un poste de travail ou une application utilisée sur le réseau.
- $I = \{I_1, I_2, ..., I_K\}$ est un ensemble de nombres réels représentant l'ensemble des menaces correspondantes aux vulnérabilités du système.
- L'IDS est distribué sur le réseau avec un ensemble de capteurs $S = \{s_1, s_2, ..., s_p\}$. Chaque capteur s_i est considéré comme un agent qui vérifie le trafic réseau et qui envoie des rapports à l'IDS. Chaque capteur peut capter plus d'une menace.
- $d = [d_1, d_2, ..., d_N]$ est un vecteur représentant une application qui associe à chaque capteur s_j un ou plusieurs éléments de $I \cup \{0\}$ tel que :

$$d_i(s_j) = \begin{cases} \mathrm{I}_k & \text{si le capteur } s_j \text{ détecte une possible intrusion ou anomalie } I_k \\ 0 & \text{sinon} \end{cases}$$

- Une matrice A qui décrit la relation entre les capteurs et les sous systèmes est définie comme suit :

$$A_{i,j} = \begin{cases} 1 & \text{si le capteur } s_j \text{ vérifie le sous système } t_i \\ 0 & \text{sinon} \end{cases}$$

- $f : I \cup \{0\} \to R^+ \cup \{0\}$ est une fonction qui associe à chaque élément de $I \cup \{0\}$ un nombre réel, sachant que $f(0) = 0$.

  Remarques :

  1. Si $f(I_m) < f(I_n)$ alors la menace I_n est plus dangeureuse que la menace I_m.
  2. La fonction f peut être appliquée aux éléments du vecteur d sans altérer les images

dans l'ensemble d'arrivée, puisque $d_i(s_j) \in I \cup \{0\}$.

- $L = \{l_1, l_2, ..., l_L\}$ est l'ensemble des niveaux de sécurité. À chaque niveau l_i correspond un seuil de sécurité m_i. Le niveau de sécurité de l'IDS est défini par :

$$L = \begin{cases} \mathrm{l}_1 & \text{si } \sum_{i=1}^{N} f(d_i) < m_1 \\ \mathrm{l}_j & \text{si } m_{j-1} \leq \sum_{i=1}^{N} f(d_i) < m_j \\ \mathrm{l}_L & \text{si } \sum_{i=1}^{N} f(d_i) > m_L \end{cases}$$

Donc, pour jouer, l'IDS et l'intrus ont chacun deux choix : l'intrus peut attaquer un ou plusieurs sous systèmes $t_i, 1 \leq i \leq M$, mais il peut aussi rester inactif (pas d'attaque). Pour facilité la notation, l'attaque porte toujours le nom du sous système destinataire.
En conclusion, l'intrus à deux stratégies, *attaquer* ou *ne pas attaquer*. Les auteurs ont associé à chaque stratégie une probabilité de réalisation. L'intrus attaque le sous système t_i avec une probabilité p et il ne l'attaque pas avec une probabilité $1 - p$. De son côté, l'IDS peut réagir aux alertes émises par les capteurs avec une probabilité q, comme il peut rester inactif face à ces alertes avec une probabilité $1 - q$. Le choix de la valeur de q est directement lié au niveau de sécurité de chaque alerte. Donc, pour les deux joueurs, le jeu consiste dans le choix des meilleures probabilités pour chacun d'entre eux.
Par la suite et en se basant sur ces résultats les auteurs ont considéré le cas ou l'attaque se fait sur plusieurs paquets [2]. Ils ont aussi utilisé les chaînes de Markov pour modéliser les alertes venant des capteurs tel que décrit dans [3].

Dans [21], K. Murali et T.V. Lakshman partent du principe que la détection d'un paquet permet de détecter l'attaque. Ils définissent ensuite un jeu entre l'intrus et l'IDS où la stratégie de l'intrus consiste à choisir un chemin pour atteindre la cible et l'IDS vérifie le trafic réseau en échantillonnant des paquets traversant certains arcs du réseau. Ils considèrent le réseau comme un graphe orienté $G(N, E)$ où :

- N et E sont respectivement l'ensemble des nœuds et l'ensemble des arcs.
- c_e est la capacité de l'arc e.
- f_e est le flux traversant l'arc e.
- s_e est le taux d'échantillonnage sur l'arc e.
- $p_e = \dfrac{s_e}{f_e}$ est la probabilité d'échantillonnage sur l'arc e.
- $U = \{p : \sum_{e \in E} p_e f_e \leq B\}$ est l'ensemble des probabilités qui respectent le seuil d'échantillonnage.

- B est le seuil d'échantillonnage, avec $\sum_{e \in E} p_e f_e \leq B$. Cela contraint l'IDS à n'echantillonner que B paquets par seconde sur tout le réseau.
- Le choix des arcs à échantillonner se fait à l'aide de l'algorithme du *minimumcut* en considérant les flux f_e.
- $q(P)$ correspond à la probabilité du choix du chemin P par l'intrus.
- P_s^t est l'ensemble des chemins menant du nœud attaquant s au nœud cible t.
- $V = \{q : \sum_{P \in P_s^t} q(P) = 1\}$ est l'ensemble représentant la distribution de la probabilité q relativement à P_s^t

La probabilité de détection est donc donnée par la formule suivante :

$$\sum_{P \in P_s^t} q(P) \sum_{e \in P} p_e$$

Cette probabilité va faire l'objet d'une maximisation pour l'IDS et d'une minimisation pour l'intrus.

Le résultat du jeu est donnée par :

$$\theta = \min_{q \in V} \max_{p \in U} \sum_{P \in P_s^t} q(P) \sum_{e \in E} p_e$$

Pour un $q \in V$ fixe, les auteurs considèrent le problème suivant :

$$max \sum_{e \in E} \left[\sum_{p \in P_s^t : P \ni e} q(P) \right] p_e$$

$$\sum_{e \in E} f_e p_e \leq B$$

$$p_e \geq 0$$

En associant une variable duale λ avec le seuil d'échantillonnage B, les auteurs ont défini le problème dual suivant :

$$\min B\lambda$$

$$f_e \lambda \geq \sum_{p \in P_s^t : P \ni e} q(P) \forall e \in E$$

$$\lambda \geq 0$$

En interprétant $q(P)$ comme étant le flux sur le chemin P, la contrainte $\sum_{p \in P_s^t : P \ni e} q(P) \leq f_e \lambda$ restreint la valeur du flux sur l'arc e à être inférieur ou égale à $f_e \lambda$. Cela veut dire que $f_e \lambda$ est égal à la capacité c_e de l'arc e. La contrainte $\sum_{p \in P_s^t} q(P) = 1$ oblige à avoir un flux égal à 1 entre les nœuds s et t. Pour que cela soit possible il faut avoir la plus petite valeur de λ. Pour solutionner le problème les auteurs proposent de :

- déterminer le flux maximal $M_{st}(f)$ entre les nœuds s et t en considérant que f_e est la capacité de l'arc e,
- prendre $\lambda = M_{st}(f)^{-1}$,
- avoir $\theta = BM_{st}(f)^{-1}$.

À partir de là les auteurs ont déterminé les stratégies des deux joueurs.
D'un côté l'intrus calcule le flux maximal $M_{st}(f)$. Il le décompose en flux $m_1, m_2, ..., m_l$ entre les éléments de l'ensemble P_s^t des chemins entre les nœuds s et t, sachant que $\sum_{i=1}^{l} m_i = M_{st}(f)$. Il envoie un paquet malicieux sur le chemin P_i avec une probabilité égale à $m_i M_{st}(f)^{-1}$.
De l'autre côté l'IDS calcule le flux maximal $M_{st}(f)$. Il détermine l'ensemble $\{e_1, e_2, ..., e_r\}$ des arcs qui compose la coupure minimale (*minimum cut*) sachant que $\sum_{i=1}^{r} f_i = M_{st}(f)$. Il échantillonne l'arc e_i avec un taux $s_{e_i} = Bf_i M_{st}(f)^{-1}$.

Dans [18 et 19] M. Mehrandish et al. ont utilisé le même principe pour traiter dans le premier temps le cas où la détection se fait sur plusieurs paquets [18]. Dans un deuxième temps, ils se sont intéressés au cas où l'attaque se fait avec une coalition d'attaquants et que la détection se fait sur plusieurs paquets [19].

N. Marchang et R. Tripathi, de leur côté, ont défini un jeu non coopératif et à somme non nulle entre l'IDS et l'intrus dans un réseau ad hoc mobile(MANET)[17]. Dans leur optique, il considère les hypothèses suivantes :

- L'IDS est présent sur tous les nœuds du réseau.
- L'IDS n'a pas besoin d'être actif durant toute la durée d'une session du réseau.
- L'intrus, non plus, n'a pas besoin d'attaquer durant toute la durée d'une session du réseau.
- M est le gain de l'IDS lorsqu'il détecte une intrusion.
- L est la perte de l'IDS lorsqu'il ne détecte pas une intrusion avec $M \geq L$.
- N est la perte de l'IDS lorsqu'il détecte une fausse intrusion.
- C_d représente le coût de l'activité de l'IDS avec $L > C_d$.
- C_a représente le coût de l'activité de l'intrus avec $L > C_a$.

Les auteurs présentent un modèle de jeu qui montre l'interaction entre l'IDS et l'intrus durant une session réseau qui dure un temps T. Pendant cette période de temps l'IDS peut surveiller le réseau durant $t\%$ du temps T, et l'intrus peut attaquer durant $s\%$ du temps T. Dans la modélisation de leur jeu ils attribuent deux stratégies pour chacun des deux joueurs. Pour l'IDS, l'ensemble des deux stratégies est noté par S_d =(*actif t%,passif*). De la même façon, ils représentent l'ensemble des deux stratégies de l'intrus par S_a =(*attaque s%,n'attaque pas*).

Dans leur recherche, les auteurs ont considéré deux types d'IDS ; l'IDS *parfait* qui est supposé avoir un taux de détection "a" égal à 100% et le taux de faux positif "b" égal à 0%. De l'autre côté l'IDS imparfait avec un taux de détection "a" et un taux de faux positif "b" compris entre 0% et 100%. Les figures 2.9, 2.10. 2.11 et 2.12 montrent les matrices de gain des joueurs pour les deux types d'IDS.

IDS/Intrus	*attaque s%*	*n'attaque pas*
actif t%	$tsM - (1-t)sL - tC_d$	$-tC_d$
passif	$-sL$	0

FIGURE 2.9 – Matrice de gain de l'IDS *parfait*

IDS/Intrus	*attaque s%*	*n'attaque pas*
actif t%	$-tsM + (1-t)sL - sC_a$	0
passif	$sL - sC_a$	0

FIGURE 2.10 – Matrice de gain de l'intrus (*IDS parfait*)

IDS/Intrus	*attaque s%*	*n'attaque pas*
actif t%	$atsM - (1-a)tsM - (1-t)sL - tC_d$	$-btN - tC_d$
passif	$-sL$	0

FIGURE 2.11 – Matrice de gain de l'IDS *imparfait*

IDS/Intrus	*attaque s%*	*n'attaque pas*
actif t%	$-atsM + (1-a)tsM + (1-t)sL - sC_a$	0
passif	$sL - sC_a$	0

FIGURE 2.12 – Matrice de gain de l'intrus (*IDS imparfait*)

La solution du jeu est un équilibre de Nash en stratégies mixtes. En effet, en supposant que p et q sont respectivement la probabilité que l'IDS choisisse d'être actif et la probabilité que l'intrus choisisse d'attaquer, les auteurs sont arrivés aux résultats suivants :

- Pour un IDS *parfait* : $p^* = \dfrac{L - C_a}{t(M+L)}$ et $q^* = \dfrac{C_d}{s(M+L)}$.
- Pour un IDS *imparfait* : $p^* = \dfrac{L - C_a}{t(2aM - M + L)}$ et $q^* = \dfrac{bN + C_d}{2asM - sM + sL + bN}$.

Chapitre 3

Présentation du jeu

3.1 Introduction

Notre travail vise à aporter une contribution dans le domaine de la détection d'intrusion. Cette contribution consiste en une solution qui permet à l'IDS de chercher la meilleure stratégie pour ses activités de vérification. Cette stratégie consiste dans le choix d'une fréquence d'activité qui permet d'optimiser le gain de l'IDS dans les conditions défavorables. Pour ce faire, nous proposons de modéliser le problème de la détection d'intrusion en utilisant la théorie des jeux selon un modèle probabiliste.

Nous considérons un jeu avec deux protagonistes ; d'un côté le système de détection d'intrusion(IDS) et de l'autre côté l'intrus. L'IDS cherche à débusquer les attaques de l'intrus pour protéger le réseau. Par ailleurs, l'intrus essaye d'atteindre sa cible et ainsi accomplir son forfait. En détection d'intrusion, l'attaque se compose de plusieurs paquets. L'intrus atteint son objectif lorsque tous les paquets arrivent à la cible. Par contre si l'IDS arrive à intercepter un certain nombre de ces paquets, l'attaque est alors débusquée.

Dans la réalité les deux adversaires adoptent des stratégies à différents niveaux et de différentes manières pour atteindre leurs objectifs. Nous avons vu, dans la revue de littérature, que les travaux qui traitent de la problèmatique l'aborde selon différents aspects.

Dans notre recherche nous considérons uniquement l'aspect qui concerne l'activité et la passivité de l'IDS et de l'intrus. Pour l'intrus, nous dirons qu'il est actif(A) lorsqu'il envoie un paquet intrusif dans le réseau, autrement il est passif(P). Pour l'IDS, nous dirons qu'il est actif quant il vérifie le trafic réseau et il est passif(P) dans le cas contraire. Nous considérons dans notre

travail que si l'intrus est actif en même temps que l'IDS, ce dernier intercepte le paquet intrusif. Par ailleurs nous associons à la stratégie de chaque joueur une probabilité qui représente le taux d'activité du joueur. En effet, nous supposons que :

- l'IDS est actif avec une probabilité p et est passif avec une probabilité $1-p$,
- l'intrus est actif avec une probabilité q et est passif avec une probabilité $1-q$.

3.2 Présentation du modèle

Nous considérons un jeu $J(n, a, b, g, c)$ où :

- n est la taille du jeu. Cela signifie qu'après n coups, de part et d'autre, on arrive à l'issue d'une partie. Donc si l'intrus n'arrive pas à envoyer tous les paquets formant l'attaque après n coups, il perd la partie et n'atteint pas son objectif même si l'IDS n'intercepte aucun paquet intrusif,
- a représente la taille d'une attaque en nombre de paquets, avec $a \leq n$,
- b correspond au nombre minimal de paquets que l'IDS doit intercepter pour détecter une intrusion, avec $b \leq a$,
- g est le gain de l'IDS. Il le perçoit soit en interceptant b paquets parmis les a, soit que l'intrus n'arrive pas à envoyer les a paquets après n coups. Il est à noter dans le dernier cas que tant que l'IDS n'a pas encore détecté b paquets, l'intrus fait quand même passer ses paquets même lorsque les deux joueurs sont actifs en même temps,
- c est le coût d'une activité de vérification unitaire de l'IDS.

Chaque étape du jeu peut avoir la configuration $\frac{A}{A}$, $\frac{A}{P}$, $\frac{P}{A}$, $\frac{P}{P}$, où le numérateur correspond à l'activité ou la passivité de l'intrus et le dénominateur correspond à l'activité ou la passivité de l'IDS.
L'IDS marque un point lorsque le résultat est $\frac{A}{A}$(paquet détecté) et l'intrus marque un point lorsque le résultat est $\frac{A}{P}$ (paquet non détecté).
L'intrus gagne lorsqu'il totalise a points en au plus n coups avant que l'IDS ne totalise b points. L'IDS gagne lorsqu'il totalise b points ou lorsque n coups ont eu lieu avant que l'intrus ne totalise a points. Pour représenter le gain de l'IDS, nous définissons les trois fonctions suivantes :

- G est la fonction du gain brut espéré qui associe à chaque couple (p, q) le gain brut espéré de l'IDS. Ce gain brut espéré s'obtient par le produit de la probabilité du gain p_g par la valeur

g du gain de l'IDS lorsqu'il intercepte une attaque.

$$G : [0,1] \times [0,1] \to R$$
$$G(p,q) = p_g * g$$

– C est la fonction du coût espéré qui associe à chaque couple (p,q) le coût espéré pour l'activité de l'IDS. Il s'obtient par le produit du nombre espéré des activités de l'IDS E_{Acids} par le coût d'une activité de vérification unitaire c.

$$C : [0,1] \times [0,1] \to R$$
$$C(p,q) = E_{Acids} * c$$

Dans la suite de ce rapport nous considérons que $c = 1$ et nous noterons le jeu par $J(n,a,b,g)$ au lieu de $J(n,a,b,g,c)$

– Pour représenter le gain de l'IDS, nous définissons la fonction du gain net espéré V qui associe à chaque couple (p,q) le gain net espéré de l'IDS. La valeur du gain net espéré de l'IDS s'obtient en soustrayant le coût du gain espéré du gain brut espéré.

$$V : [0,1] \times [0,1] \to R$$
$$V(p,q) = G(p,q) - C(p,q)$$

Il est évident que l'issue du jeu se décide en fonction de la valeur de $V(p,q)$. En effet, l'IDS tentera de la maximiser, par contre, l'intrus s'efforcera à la minimiser. On suppose que l'intrus sera actif avec une probabilité d'activité qui minimisera le gain net espéré de l'IDS, il cherchera donc le pire cas pour l'IDS quelque soit la valeur de p. De l'autre côté, sachant que son adverssaire le met dans les situations les plus défavorables, l'IDS sera actif avec une probabilité d'activité qui lui permettra d'atteindre le plus haut gain en pire cas. Pour résoudre le jeu, il faut d'abord trouver les valeurs, $q^*(p)$, de la probabilité de l'activité de l'intrus qui minimise le gain de l'IDS pour chaque valeur de p , c'est à dire déterminer la valeur $q^*(p)$ telle que pour chaque valeur de p

$$V(p,q^*(p)) = min\{V(p,q) : q \in [0,1]\}.$$

Ensuite, nous trouverons la valeur p^* de la probabilité de l'activité de l'IDS qui maximise le gain pour chaque valeur de $q^*(p)$, c'est à dire déterminer p^* telle que pour chaque valeur de $q^*(p)$

$$V(p^*, q^*(p^*)) = max\{V(p, q^*(p)) : p \in [0, 1]\}$$

Cette valeur de p^* est la probabilité que doit utiliser l'IDS pour maximiser son gain en pire cas.

Pour calculer le coût de l'IDS nous utiliserons les notations suivantes :

1. $P_k(J)$ signifie que le joueur J est passif au $k^{ème}$ coup.
2. $A_k(J)$ signifie que le joueur J est actif au $k^{ème}$ coup.
3. $C^{A_k(Int),j}_{k,i,A_k(IDS)}$: le coût de l'IDS lorsque :
   - la partie se termine en exactement k coups,
   - l'IDS est actif i fois,
   - l'IDS est actif au $k^{ème}$ coup,
   - l'intrus est actif au $k^{ème}$ coup
   - et l'intrus ne se fait pas attraper j fois dans les $(k-1)$ premiers coups.
4. $C^{A_k(Int)}_{k,i,A_k(IDS)}$: le coût de l'IDS lorsque :
   - la partie se termine en exactement k coups,
   - l'IDS est actif i fois,
   - l'IDS est actif au $k^{ème}$ coup
   - et l'intrus est actif au $k^{ème}$ coup
5. $C_{k,i,P_k(IDS)}$: le coût de l'IDS lorsque :
   - la partie se termine en exactement k coups,
   - l'IDS est actif i fois,
   - l'IDS est passif au $k^{ème}$ coup
6. $C_{k,i}$: le coût de l'IDS lorsque :
   - la partie se termine en exactement k coups,
   - et l'IDS est actif i fois.

Remarque : Lorsqu'on omet de spécifier $A_k(J)$ et $P_k(J)$ dans une formule, cela signifie que le joueur J peut être actif ou passif

Dans la suite de ce travail, nous allons consacrer un chapitre pour l'étude du jeu $J(n, a, 1, g)$ où l'IDS n'a qu'à intercepter un seul paquet pour débusquer l'attaque. Nous allons ensuite, étudier le jeu $J(n, n, n, g)$ où la taille de l'attaque, la taille du jeu ainsi que le nombre de paquets nécessaires pour débusquer une attaque sont égaux.

Chapitre 4

Résolution du jeu $J(n, a, 1, g)$

4.1 Introduction

Dans ce chapitre nous allons considérer le jeu $J(n, a, 1, g)$ qui peut durer n coups avec $n \geq 1$. L'attaque doit contenir a paquets, sachant que $a \leq n$. Le gain de l'IDS lorsqu'il intercepte une attaque est $g \geq 0$. L'IDS gagne s'il intercepte un seul paquet intrusif avant la fin du jeu ou si la partie se termine alors que l'intrus n'a pas réussit à passer les a paquets. Résoudre ce jeu veut dire déterminer la valeur p^* de la probabilité d'activité de l'IDS qui lui assure le meilleur gain en pire cas. Pour ce faire nous allons procéder par étapes. Dans un premier temps, nous allons déterminer la formule générale permettant de calculer le gain net espéré de l'IDS. Nous allons, ensuite, résoudre analytiquement le jeu pour des valeurs déterminées de n et de a. Enfin, nous allons généraliser pour certaines grandes valeurs de ces paramètres.

4.2 Formule générale calculant $V(p, q)$

Dans cette section nous allons déterminer la formule générale qui nous permettra de calculer le gain net espéré de l'IDS $V(p, q)$. Pour se faire, nous allons considérer un jeu $J(n, a, 1, g)$ avec n, a, g quelconques. Le nombre minimal de paquets, b, que doit intercepter l'IDS pour détecter l'attaque est fixé à 1. Sachant que le gain net espéré se calcule par la différence entre le gain brut espéré et le coût espéré, $V(p, q) = G(p, q) - C(p, q)$, nous allons d'abord les déterminer un par un, ensuite faire la différence.

4.2.1 Formule de calcul du gain brut espéré de l'IDS $G(p, q)$

Pour déterminer la formule de calcul du gain brut espéré $G(p, q)$, il suffit de calculer la probabilité du gain de l'IDS. Pour ce faire, nous allons procéder comme suit :
Nous allons d'abord calculer la probabilité de la perte de l'IDS p_{perte}. Par la suite il suffit de prendre le complémentaire à 1 pour avoir la probabilité du gain. En d'autre terme $p_g = 1 - p_{perte}$. Pour le jeu $J(n, a, 1, g)$, nous savons que la partie peut se terminer soit au dernier coup n, soit avant le $n^{ième}$ coup. En supposant que l'IDS perd au $k^{ième}$ coup avec la probabilité p^k_{perte}, la probabilité de perte de l'IDS p_{perte} sera égale à la somme des probabilités de la perte du $a^{ième}$ coup au $n^{ième}$ coup. L'IDS ne peut évidement pas perdre avant le $a^{ième}$ coup, car la taille de l'attaque est de a. En d'autre termes $p_{perte} = \sum_{k=a}^{n} p^k_{perte}$. Ceci dit, la perte de l'IDS au $k^{ième}$ coup est conditionnée par les trois points suivants :

1. La partie se termine avec la combinaison $\frac{A}{P}$, c'est à dire que l'intrus est actif par contre l'IDS est passif. Cette condition se réalise avec la probabilité $(1 - p)q$.
2. Pour accomplir sont attaque il ne reste à l'intrus qu'un seul paquet intrusif à envoyer au $k^{ième}$ coup. L'intrus doit donc avoir été actif $(a - 1)$ fois dans les $(k - 1)$ premiers coups.
3. Il ne doit pas y avoir de combinaison $\frac{A}{A}$ dans les $(k - 1)$ premiers coups, sinon la partie serait terminée avant le $k^{ième}$ coup avec l'IDS gagnant.

La deuxième et la troisième conditions se réalisent avec la probabilité $\binom{k-1}{a-1} q^{a-1}(1-p)^{a-1}(1-q)^{k-a}$. Cela veut dire que l'IDS perd, au $k^{ième}$ coup exactement, avec la probabilité
$p^k_{perte} = \binom{k-1}{a-1}(1-p)^a q^a (1-q)^{k-a}$. En faisant la sommation de toutes les probabilités de la perte à partir du $a^{ième}$ coup jusqu'au $n^{ième}$ coup on obtiendra, donc, la probabilité de la perte de l'IDS $p_{perte} = (1-p)^a q^a \sum_{k=a}^{n} \binom{k-1}{a-1}(1-q)^{k-a}$.
En conclusion, nous dirons que :

- la probabilité du gain de l'IDS est : $p_g = 1 - (1-p)^a q^a \sum_{k=a}^{n} \binom{n-1}{a-1}(1-q)^{k-a}$,
- la formule du gain brut espéré de l'IDS est donnée par :

$$G(p, q) = \left(1 - (1-p)^a q^a \sum_{k=a}^{n} \binom{k-1}{a-1}(1-q)^{k-a}\right) g \tag{4.1}$$

4.2.2 Formule de calcul du coût espéré de l'IDS

Le coût espéré de l'IDS dépend du nombre de ses activités durant le jeu. Dans le cas où, par exemple, la partie s'arrête au $k^{ième}$ coup, l'IDS ne peut pas payer plus que k fois. Par contre, il peut payer moins, s'il n'a pas été tout le temps actif. Pour calculer, donc, ce coût, il faut prendre

en considération le nombre de coups que dure la partie ainsi que le nombre de fois où l'IDS est actif. Dans une partie à k coups($k \leq n$), l'IDS peut être actif i fois avec $0 < i \leq k$. Nous allons déterminer les coûts dans le cas où la partie finisse avant le $n^{ième}$ coup et dans le cas où elle finisse au $n^{ième}$ coup exactement.

1. Dans le cas où la partie se termine au $k^{ième}$ coup avec $k < n$, l'intrus doit être actif au dernier coup. La fin sera, donc, caractérisée de deux façons. La première, lorsque l'IDS est actif et la deuxième, lorsqu'il est passif. On peut, alors, dire qu'une partie peut se terminer soit avec la combinaison $\frac{A}{A}$, soit avec la combinaison $\frac{A}{P}$.

   – Si la partie se termine avec la combinaison $\frac{A}{A}$, cela veut dire que l'intrus ainsi que l'IDS sont actifs au dernier coup. Pour que cela se réalise, il faut que les conditions suivantes soient vérifiées :

   (a) Il n'y a pas eu de combinaison $\frac{A}{A}$ dans les $(k-1)$ premiers coups pour éviter que la partie ne se termine avant le $k^{ième}$ coup,

   (b) l'IDS doit être actif $(i-1)$ fois dans les $(k-1)$ premiers coups, puisqu'il est actif à la fin de la partie et qu'il ne peut l'être que i fois exactement,

   (c) l'intrus peut être actif au plus $(a-1)$ fois dans les $(k-1)$ premiers coups sinon la partie serait terminée avant le $k^{ième}$ coup. On suppose donc que l'intrus est actif j fois, sachant que $j \leq a-1$.

   En conclusion, nous dirons que :

   – les trois conditions précédentes se réalisent avec la probabilité

$$\binom{k-1}{i-1} p^{i-1}(1-q)^{i-1}(1-p)^{k-i} \sum_{j=0}^{a-1} \binom{k-i}{j} q^j (1-q)^{k-i-j} pq$$

   – le coût espéré de l'IDS lorsque la partie se termine au $k^{ième}$ coup, exactement, quand l'intrus et l'IDS sont actifs est :

$$C^{A_k(Int)}_{k,i,A_k(IDS)} = i\binom{k-1}{i-1} p^i (1-p)^{k-i} q(1-q)^{i-1} \sum_{j=0}^{a-1} \binom{k-i}{j} q^j (1-q)^{k-i-j} \qquad (4.2)$$

– Si la partie se termine avec la combinaison $\frac{A}{P}$, cela veut dire qu'au dernier coup l'intrus est actif mais l'IDS est passif. Les conditions suivantes sont nécessaires pour que cela se produise :

  (a) Il n'y a pas eu de combinaison $\frac{A}{A}$ dans les $(k-1)$ premiers coups sinon la partie serait terminée avant le $k^{ième}$ coup,

  (b) l'IDS doit être actif i fois dans les $(k-1)$ premiers coups,

  (c) l'intrus doit être actif $(a-1)$ fois dans les $(k-1)$ premiers coups pour qu'il puisse faire passer tous ses paquets avec son action au $k^{ième}$ coup.

  En conclusion, nous dirons que :

  – les trois conditions précédentes se réalisent avec la probabilité

$$\binom{k-1}{i}p^i(1-q)^i(1-p)^{k-i-1}\binom{k-i-1}{a-1}q^{a-1}(1-q)^{k-i-a}(1-p)q$$

  – le coût espéré de l'IDS lorsque la partie se termine au $k^{ième}$ coup, exactement, quand l'intrus est actif mais l'IDS passif est :

$$C^{A_k(Int)}_{k,i,P_k(IDS)} = i\binom{k-1}{i}p^i(1-p)^{k-i}\binom{k-i-1}{a-1}q^a(1-q)^{k-a} \tag{4.3}$$

Le coût de l'IDS pour le jeu $J(n,a,1,g)$ lorsque la partie se termine, exactement, au $k^{ième}$ coup (avec $k<n$) est donc :

$$C_{k,i} = C^{A_k(Int)}_{k,i,A_k(IDS)} + C^{A_k(Int)}_{k,i,P_k(IDS)} \tag{4.4}$$

Le calcul du coût C_k lorsque la partie se termine au $k^{ième}$ coup, avec $k<n$, s'obtient par la double sommation sur k et i appliquée à $C_{k,i}$. En d'autres termes

$$C_k = \sum_{k=1}^{n-1}\sum_{i=1}^{k} C_{k,i} \tag{4.5}$$

Par les équations (4.4) et (4.5) nous avons :

$$C_k = \sum_{k=1}^{n-1}\sum_{i=1}^{k} C^{A_k(Int)}_{k,i,A_k(IDS)} + \sum_{k=1}^{n-1}\sum_{i=1}^{k} C^{A_k(Int)}_{k,i,P_k(IDS)} \tag{4.6}$$

2. Dans le cas où la partie se termine au $n^{ième}$ coup, la fin sera caractérisée de quatre façons différentes.
   - Si la partie se termine avec la combinaison $\frac{A}{A}$, cela veut dire que les deux joueurs sont actifs au dernier coup. Pour que cela soit possible il faut que les conditions suivantes soient vérifiées :
     (a) Il n'y a pas eu de combinaison $\frac{A}{A}$ dans les $(n-1)$ premiers coups sinon la partie serait terminée avant le $n^{ième}$ coup,
     (b) l'IDS doit être actif $(i-1)$ fois dans les $(n-1)$ premiers coups,
     (c) l'intrus peut être actif au plus $(a-1)$ fois dans les $(n-1)$ premiers coups sinon la partie serait terminée avant le $n^{ième}$ coup. On suppose donc que l'intrus est actif j fois, sachant que $j \leq a-1$.

     En conclusion, nous dirons que :
     - les trois conditions précédentes se réalisent avec la probabilité

$$\binom{n-1}{i-1}p^{i-1}(1-q)^{i-1}(1-p)^{n-i}\sum_{j=0}^{a-1}\binom{n-i}{j}q^j(1-q)^{n-i-j}pq$$

     - le coût espéré de l'IDS lorsque la partie se termine au $n^{ième}$ coup, exactement, quand l'intrus et l'IDS sont actifs est :

$$C^{A_n(Int)}_{n,i,A_n(IDS)} = i\binom{n-1}{i-1}p^i(1-p)^{n-i}q(1-q)^{i-1}\sum_{j=0}^{a-1}\binom{n-i}{j}q^j(1-q)^{n-i-j} \qquad (4.7)$$

   - Si la partie se termine avec la combinaison $\frac{A}{P}$, cela veut dire que l'intrus est actif par contre, l'IDS est passif. Dans ce cas les conditions suivantes doivent être vérifiées :
     (a) Il n'y a pas eu de combinaison $\frac{A}{A}$ dans les $(n-1)$ premiers coups sinon la partie serait terminée avant le $n^{ième}$ coup,
     (b) l'IDS doit être actif i fois dans les $(n-1)$ premiers coups,
     (c) l'intrus peut être actif au plus $(a-1)$ fois dans les $(n-1)$ premiers coups sinon la partie serait terminée avant le $n^{ième}$ coup. On suppose donc que l'intrus est actif j fois, sachant que $j \leq a-1$.

     En conclusion, nous dirons que :
     - les trois conditions précédentes se réalisent avec la probabilité

$$\binom{n-1}{i}p^i(1-q)^i(1-p)^{n-i-1}\sum_{j=0}^{a-1}\binom{n-i-1}{j}q^j(1-q)^{n-i-j-1}(1-p)q$$

– le coût espéré de l'IDS lorsque la partie se termine au $n^{ième}$ coup, exactement, quand l'intrus est actif et l'IDS pactif est :

$$C^{A_n(Int)}_{n,i,P_n(IDS)} = i\binom{n-1}{i}p^i(1-p)^{n-i}q(1-q)^i\sum_{j=0}^{a-1}\binom{n-i-1}{j}q^j(1-q)^{n-i-j-1} \quad (4.8)$$

– Si la partie se termine avec la combinaison $\frac{P}{A}$, cela veut dire que l'intrus est passif mais l'IDS pas contre, il est acti. Dans ce cas les conditions suivantes doivent être vérifiées :

(a) Il n'y a pas eu de combinaison $\frac{A}{A}$ dans les $(n-1)$ premiers coups sinon la partie serait terminée avant le $n^{ième}$ coup,

(b) l'IDS doit être actif $(i-1)$ fois dans les $(n-1)$ premiers coups,

(c) l'intrus peut être actif au plus $(a-1)$ fois dans les $(n-1)$ premiers coups sinon la partie serait terminée avant le $n^{ième}$ coup. On suppose donc que l'intrus est actif j fois, sachant que $j \leq a-1$.

En conclusion nous dirons que :

– les trois conditions précédentes se réalisent avec la probabilité

$$\tbinom{n-1}{i-1}p^{i-1}(1-q)^{i-1}(1-p)^{n-i}\textstyle\sum_{j=0}^{a-1}\tbinom{n-i}{j}q^j(1-q)^{n-i-j}p(1-q)$$

– le coût espéré de l'IDS lorsque la partie se termine au $n^{ième}$ coup, exactement, quand l'intrus passif et l'IDS actifs est :

$$C^{P_n(Int)}_{n,i,A_n(IDS)} = i\binom{n-1}{i-1}p^i(1-p)^{n-i}(1-q)^i\sum_{j=0}^{a-1}\binom{n-i}{j}q^j(1-q)^{n-i-j} \quad (4.9)$$

– Si la partie se termine avec la combinaison $\frac{P}{P}$, cela veut dire que les deux joueurs sont tous les deux passifs. Dans ce cas, les conditions suivantes doivent être vérifiées :

(a) Il n'y a pas eu de combinaison $\frac{A}{A}$ dans les $(n-1)$ premiers coups sinon la partie serait terminée avant le $n^{ième}$ coup,

(b) l'IDS doit être actif i fois dans les $(n-1)$ premiers coups,

(c) l'intrus peut être actif au plus $(a-1)$ fois dans les $(n-1)$ premiers coups sinon la partie serait terminée avant le $n^{ième}$ coup. On suppose donc que l'intrus est actif j fois, sachant que $j \leq a-1$.

En conclusion, nous dirons que :

– les trois conditions précédentes se réalisent avec la probabilité

$$\binom{n-1}{i}p^i(1-q)^i(1-p)^{n-i-1}\sum_{j=0}^{a-1}\binom{n-i-1}{j}q^j(1-q)^{n-i-j-1}(1-p)(1-q)$$

– le coût espéré de l'IDS lorsque la partie se termine au $n^{ième}$ coup, exactement, quand l'intrus et l'IDS sont pactifs est :

$$C^{P_n(Int)}_{n,i,P_n(IDS)} = i\binom{n-1}{i}p^i(1-p)^{n-i}(1-q)^{i+1}\sum_{j=0}^{a-1}\binom{n-i-1}{j}q^j(1-q)^{n-i-j-1} \quad (4.10)$$

Le coût de l'IDS pour le jeu $J(n,a,1,g)$ lorsque la partie se termine au $n^{ième}$ coup est donné par :

$C_{n,i} = C^{A_n(Int),j}_{n,i,A_n(IDS)} + C^{A_n(Int),j}_{n,i,P_n(IDS)} + C^{P_n(Int),j}_{n,i,P_n(IDS)} + C^{P_n(Int),j}_{n,i,A_n(IDS)}$

Or que par les équations (4.7) et (4.9) nous avons :

$$C^{A_n(Int),j}_{n,i,A_n(IDS)} + C^{P_n(Int),j}_{n,i,A_n(IDS)} = i\binom{n-1}{i-1}p^i(1-p)^{n-i}(1-q)^{i-1}\sum_{j=0}^{a-1}\binom{n-i}{j}q^j(1-q)^{n-i-j} \quad (4.11)$$

et par les équations (4.8) et (4.10) nous avons :

$$C^{A_n(Int),j}_{n,i,P_n(IDS)} + C^{P_n(Int),j}_{n,i,P_n(IDS)} = i\binom{n-1}{i}p^i(1-p)^{n-i}(1-q)^{i}\sum_{j=0}^{a-1}\binom{n-i-1}{j}q^j(1-q)^{n-i-j-1} \quad (4.12)$$

On a donc

$$\begin{aligned} C_{n,i} &= i\binom{n-1}{i-1}p^i(1-p)^{n-i}(1-q)^{i-1}\sum_{j=0}^{a-1}\binom{n-i}{j}q^j(1-q)^{n-i-j} \\ &+ i\binom{n-1}{i}p^i(1-p)^{n-i}(1-q)^{i}\sum_{j=0}^{a-1}\binom{n-i-1}{j}q^j(1-q)^{n-i-j-1} \end{aligned} \quad (4.13)$$

Le calcul du coût C_n lorsque la partie se termine au $n^{ième}$ coup s'obtient par sommation sur i appliquée à $C_{n,i}$. En d'autres termes

$$C_n = \sum_{i=1}^{n} C_{n,i} \quad (4.14)$$

Par les équations (4.13) et (4.14) nous avons :

$$\begin{aligned} C_n = &\sum_{i=1}^{n} i\binom{n-1}{i-1}p^i(1-p)^{n-i}(1-q)^{i-1}\sum_{j=0}^{a-1}\binom{n-i}{j}q^j(1-q)^{n-i-j} \\ &+\sum_{i=1}^{n} i\binom{n-1}{i}p^i(1-p)^{n-i}(1-q)^{i}\sum_{j=0}^{a-1}\binom{n-i-1}{j}q^j(1-q)^{n-i-j-1} \end{aligned} \tag{4.15}$$

En utilisant les équations (4.6) et (4.14) nous concluons que le coût, $C(p,q) = C_k + C_n$, de l'IDS pour le jeu $J(n,a,1,g)$ est :

$$C(p,q) = \sum_{k=1}^{n-1}\sum_{i=1}^{k} C_{k,i,A_k(IDS)}^{A_k(Int)} + \sum_{k=1}^{n-1}\sum_{i=1}^{k} C_{k,i,P_k(IDS)}^{A_k(Int)} + \sum_{i=1}^{n} C_{n,i} \tag{4.16}$$

4.2.3 Calcul du gain net espéré

Par les équations (4.1),(4.16) et la fonction du gain net espéré définie au paragraphe 3.2, nous obtenons la formule générale de calcul de $V(p,q)$

$$\begin{aligned} V(p,q) = &\left(1-(1-p)^a q^a \sum_{k=a}^{n}\binom{k-1}{a-1}(1-q)^{k-a}\right) g \\ &-\sum_{k=1}^{n-1}\sum_{i=1}^{k} C_{k,i,A_k(IDS)}^{A_k(Int)} - \sum_{k=1}^{n-1}\sum_{i=1}^{k} C_{k,i,P_k(IDS)}^{A_k(Int)} - \sum_{i=1}^{n} C_{n,i} \end{aligned} \tag{4.17}$$

Remarque

Dans le cas particulier du jeu $J(n,a,1,g)$ où $n = a = 1$, le calcul du coût ne prend en considération que le cas où le jeu se termine au $n^{ième}$ coup.

4.3 Résolution du jeu $J(1,1,1,g)$

Dans cette section nous allons déterminer la valeur p^* de la probabilité d'activité de l'IDS qui lui assure le meilleur gain en pire cas pour un jeu où :

– la taille n est égale à 1,

– la taille a de l'attaque est de 1,

– une intrusion est détecté si l'IDS intercepte un seul paquet ($b = 1$),

– le gain g de l'IDS lorsqu'il intercepte une attaque est supérieur à zéro

Dans ces conditions où $n = 1$, si l'intrus n'est pas actif, il perd la partie après le premier coup. L'intrus n'a donc d'autres choix que d'être actif. De la même manière, vu que le jeu ne dure qu'un seul coup et que l'attaque n'est que sur un seul paquet, si l'IDS reste passif, il risque de perdre la partie. Son meilleur choix dépend de la grandeur du gain g par rapport

au coût unitaire.

Pour $n = 1$, l'issue d'une partie est en faveur de l'intrus uniquement s'il est actif et que l'IDS est passif comme le montre la figure 4.1.

IDS	P	P	A	A
Intrus	P	A	P	A
Vainqueur	IDS	Intrus	IDS	IDS

FIGURE 4.1 – Issues du jeu pour $n = 1$ et $a = 1$

La première observation qui découle est que $G(p, q) = (1 - q(1 - p))g$ puisque la probabilité du gain de l'IDS est égale à $1 - q(1 - p)$. La deuxième observation c'est le coût espéré $C(p) = p$, puisque $n = 1$. De ces deux constatations découle la valeur du gain net espéré :

$$V(p, q) = (1 - q(1 - p))g - p$$

En effet, si nous remplaçons n, a, et b par leurs valeurs correspondantes du jeu $J(1, 1, 1, g)$ dans la formule générale du gain net espéré, nous obtenons la même expression pour le gain net espéré. On remarque que quelque soit la probabilité d'activité de l'IDS, la valeur du gain net espéré $V(p, q)$ est minimale si la valeur de $q(1 - p)$ est maximale. Pour que cela soit vrai il faut que q soit égal à 1. La probabilité de l'activité de l'intrus qui minimise le gain net espéré de l'IDS est donc $q^*(p) = 1$.

En remplaçant p par $q^*(p)$ dans la formule de calcul du gain net espéré de l'IDS on aura :

$$V(p, 1) = (1 - (1 - p))g - p$$
$$V(p, 1) = pg - p$$
$$V(p, 1) = p(g - 1)$$

Puisque la valeur de $V(p, 1)$ dépend de p et de g, il faut considérer deux cas :

1^{er}cas : Lorsque $g > 1$, la valeur de $V(p, 1)$ est maximale et est égale à $g - 1$, quand $p = 1$.

$2^{ème}$cas : Lorsque $g < 1$, la valeur de $V(p, 1)$ est maximale et égale à 0, quand $p = 0$.

En conclusion, nous dirons que :

– Lorsque $g > 1$, la valeur de la probabilité d'activité de l'IDS qui lui assure le meilleur gain en pire cas est $p^* = 1$.

– Lorsque $g \leq 1$, la valeur de la probabilité d'activité de l'IDS qui lui assure le meilleur gain en pire cas est $p^* = 0$.

4.4 Résolution du jeu $J(2, 1, 1, g)$

Dans cette section nous allons déterminer la valeur p^* de la probabilité d'activité de l'IDS qui lui assure le meilleur gain en pire cas pour un jeu où :

– la taille n est égale à 2,

– la taille a de l'attaque est de 1,

– une intrusion est détectée si l'IDS intercepte un paquet ($b = 1$),

– le gain g de l'IDS lorsqu'il intercepte une attaque est supérieur ou égal à zéro

Nous allons utiliser la formule générale de calcul du gain net espéré, $V(p, q)$, vue dans le chapitre 3 pour calculer $q^*(p)$ et p^*. La formule de $V(p, q)$ est une fonction à deux inconnues (p et q). Pour résoudre ce jeu, nous allons d'abord chercher la valeur $q^*(p)$, de la probabilité d'activité de l'intrus qui minimise la valeur de $V(p, q)$ quelque soit la valeur de p. Par la suite nous allons calculer la valeur p^* de la la probabilité d'activité de l'IDS qui maximise $V(p, q^*(p))$.

Dans le cas présent, nous allons détailler le calcul du la formule du gain net espéré, $V(p, q) = G(p, q) - C(p, q)$, en utilisant la formule générale de calcul du gain net espéré de l'IDS.

Par les équations (4.1) et (4.16), nous avons :

$$G(p,q) = \left(1 - (1-p)q \sum_{k=1}^{2} \binom{k-1}{0} (1-q)^{k-1}\right) g \tag{4.18}$$

et

$$C(p,q) = \sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,A_k(IDS)}^{A_k(Int)} + \sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,P_k(IDS)}^{A_k(Int)} + \sum_{i=1}^{2} C_{2,i} \tag{4.19}$$

D'un côté le développement de l'équation (4.18) nous donne

$G(p,q) = \left(1 - (1-p)q(\binom{0}{0}(1-q)^0 + \binom{1}{0}(1-q)^1\right) g$

$= (1 - (1-p)q(1 + (1-q))) g$

$= (1 - (1-p)(q + q(1-q))) g$

$= \left(1 - (1-p)(q + q - q^2)\right)) g$

$= \big(1 - (1-p)(2q - q^2)\big)\, g$

$= \big(1 + (1-p)q^2 - 2(1-p)q\big)\, g$

En d'autres termes

$$G(p,q) = g + g(1-p)q^2 - 2g(1-p)q \tag{4.20}$$

De l'autre côté le développement de l'équation (4.19) nous donne

(a) $\sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,A_k(IDS)}^{A_k(Int)} = \binom{0}{0}p(1-p)^0 q(1-q)^0(\binom{0}{0}q^0(1-q)^0)$

En d'autres termes

$$\sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,A_k(IDS)}^{A_k(Int)} = pq,$$

(b) $\sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,P_k(IDS)}^{A_k(Int)} = \binom{0}{1}p(1-p)^0\binom{-1}{0}q(1-q)^0$

Puisque $\binom{0}{1} = 0$ cela implique

$$\sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,P_k(IDS)}^{A_k(Int)} = 0,$$

(c) $\sum_{i=1}^{2} C_{2,i} = \binom{1}{0}p(1-p)^0(1-q)^0(\binom{0}{0}q^0(1-q)) + 2\binom{1}{1}p^2(1-p)^0(1-q)(\binom{1}{0}q^0(1-q)^0) +$
$\binom{1}{1}p(1-p)(1-q)(\binom{0}{0}q^0(1-q)^0) + 2\binom{1}{2}p^2(1-p)^0(1-q)^2(\binom{-1}{0}q^0(1-q)^{-2})$

Puisque $\binom{1}{2} = 0$ on aura :
$\sum_{i=1}^{2} C_{2,i} = \binom{1}{0}p(1-p)^0(1-q)^0(\binom{0}{0}q^0(1-q)) + 2\binom{1}{1}p^2(1-p)^0(1-q)(\binom{1}{0}q^0(1-q)^0) +$
$\binom{1}{1}p(1-p)(1-q)(\binom{0}{0}q^0(1-q)^0)$

$\sum_{i=1}^{2} C_{2,i} = p(1-p)(1-q) + 2p^2(1-q) + p(1-p)(1-q)$

$\sum_{i=1}^{2} C_{2,i} = 2p - 2pq$

À partir de là, nous dirons que le coût espéré de l'IDS pour le jeu $j(2,1,1,g)$ est :

$$C(p,q) = 2p - pq \tag{4.21}$$

Les équations (4.20),(4.21) et la fonction du gain net définie au paragraphe 3.2 impliquent que

$$V(p,q) = g(1-p)q^2 - (2g(1-p) - p)q + g - 2p. \tag{4.22}$$

En considérant g comme étant une constante, la valeur du gain net espéré de l'IDS dans ce cas est une fonction à deux variables p et q comme on l'a dit plus haut, la figure suivante montre le graphique qui met en évidence le comportement de cette fonction pour un valeur de $g = 5$.

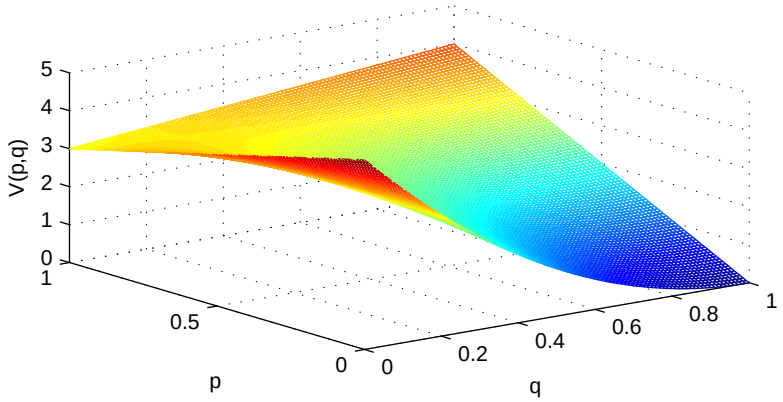

FIGURE 4.2 – Graphe de la fonction du gain net espéré de l'IDS pour $g = 5$

4.4.1 Probabilité de l'activité de l'intrus qui minimise le gain net espéré de l'IDS : $q^*(p)$

Pour calculer $q^*(p)$, nous allons étudier la fonction du gain net espéré de l'IDS par rapport à la variable q. Ceci dit, la dérivée de $V(p,q)$ par rapport à q,

$$\frac{\partial V}{\partial q} = 2g(1-p)q - (2g(1-p) - p),$$

s'annule pour

$$q = \frac{2g(1-p) - p}{2g(1-p)}.$$

De plus, la dérivée seconde de $V(p,q)$ par rapport à q,

$$\frac{\partial^2 V}{\partial^2 q} = 2g(1-p),$$

est toujours positive. Cela implique que $V(p,q)$ est minimale pour

$$q = \frac{2g(1-p) - p}{2g(1-p)}.$$

En conclusion nous dirons que la valeur $q^*(p)$ de la probabilité d'activité de l'intrus qui minimise la valeur du gain net espéré de l'IDS quelque soit la valeur de la probabilité d'activité de l'IDS p est donnée par :

$$q^*(p) = \frac{2g(1-p)-p}{2g(1-p)}$$

4.4.2 Probabilité de l'activité de l'IDS qui maximise la gain net espéré de l'IDS en pire cas : p^*

Pour calculer p^*, nous allons remplacer, dans la formule de calcul du gain net espéré de l'IDS $V(p,q)$, la valeur q par $q^*(p)$.
On aura donc $V(p,q^*(p)) = -\frac{p^2}{4g(1-p)} + (g-1)p$

La dérivée $V'(p,q^*(p)) = \frac{(4g^2-4g+1)p^2 - 2(4g^2-4g+1)p + 4g^2 - 4g}{4g(1-p)^2}$ s'annule pour :

$$p_1 = 1 - \frac{1}{\sqrt{4g^2-4g+1}} \text{ et } p_2 = 1 + \frac{1}{\sqrt{4g^2-4g+1}}$$

On remarque que $0 \leq p_1 \leq 1$ et $p_2 > 1$. Cela veut dire que p_2 ne peut pas constituer une solution pour notre problème puisque p^* doit être compris entre 0 et 1.
Pour calculer p^* on va considérer les deux cas suivants :

– Si $g \geq 1$:

Nous avons d'un côté,

$$p_1 = 1 - \frac{1}{\sqrt{4g^2-4g+1}}$$

constitue un extrémum de la fonction $V(p,q^*(p))$ et il est compris entre 0 et 1.

De l'autre côté, la dérivée seconde,

$$V''(p,q^*(p)) = -\frac{1}{2g(1-p)^3},$$

est toujours négative. Cela implique, donc, que p_1 maximise la fonction$V(p,q^*(p))$. En conclusion nous dirons que la valeur de la probabilité de l'activité de l'IDS qui maximise la fonction du gain net espéré de l'IDS est :

$$p^* = 1 - \frac{1}{\sqrt{4g^2-4g+1}}$$

Le graphe suivant montre bien la variation de $V(p,q^*(p))$ selon les valeurs de la probabilité d'activité de l'IDS.

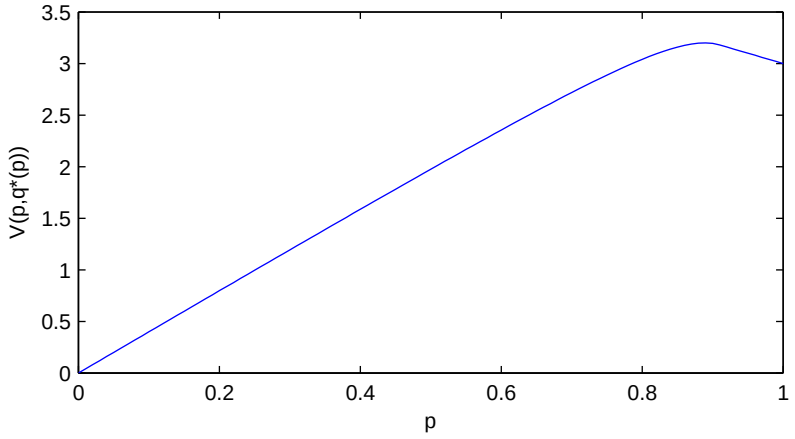

FIGURE 4.3 – Graphe de $V(p, q^*(p))$ pour $g = 5$

– Si $g < 1$:

Dans ce cas nous avons $V'(p, q^*(p)) \leq 0$, ce qui signifie que la fonction $V(p, q^*(p))$ est décroissante. Cela signifie aussi que le maximum de $V(p, q^*(p))$ est atteint pour $p = 0$ et que $q^*(0) = 1$

En conclusion nous dirons que la valeur de la probabilité de l'activité de l'IDS qui maximise sa fonction du gain net espéré est :

$$p^* = 0$$

Le graphe de variation de $V(p, q^*(p))$ confirme que pour $g < 1$, le maximum du gain net espéré de l'IDS est atteint pour $p^* = 0$.

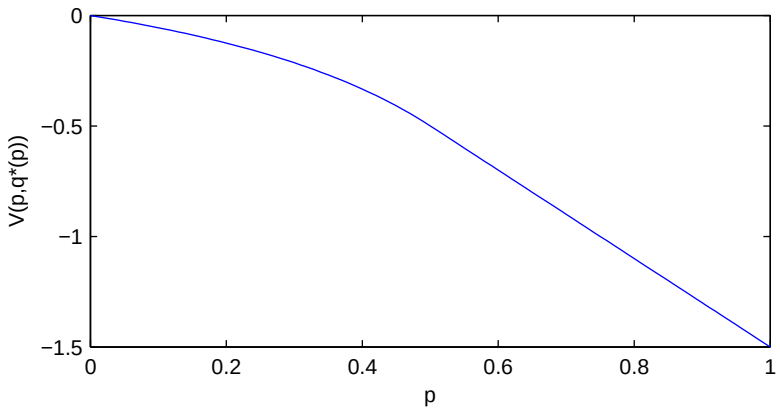

FIGURE 4.4 – Graphe de $V(p, q^*(p))$ pour $g = \frac{1}{2}$

4.5 Résolution du jeu $J(n, 1, 1, g)$

Résoudre analytiquement ce jeu pour $n \geq 3$ semble être difficile. En effet, la fonction du gain V est de degré n par rapport à la variable q et c'est cela qui rend sa résolution difficile. Juste pour $n = 3$, le gain net espéré de l'IDS est donné par

$$V(p,q) = -g(1-p)q^3 + (3g(1-p) - p)\, q^2 - 3\,(g(1-p) - p)\, q - 3p + g$$

Sa dérivée par rapport à q,

$$\frac{\partial V}{\partial q} = -3g(1-p)q^2 + 2\,(3g(1-p) - p)\, q - 3\,(g(1-p) - p)\,,$$

s'annule pour

$$q_1 = \frac{(3g(1-p) - p) + \sqrt{p(3g(1-p) + p)}}{3g(1-p)} \text{ et } q_2 = \frac{(3g(1-p) - p) - \sqrt{p(3g(1-p) + p)}}{3g(1-p)}$$

Entre ces deux valeurs de la probabilité d'activité de l'intrus c'est q_2 qui minimise le gain net espéré de l'IDS, en d'autres termes $q^*(p) = q_2$. En remplaçant la valeur q de la probabilité d'activité de l'intrus dans la fonction du gain net espéré de l'IDS par $q^*(p)$, on obtiendra une fonction de troisième degré avec certains polynômes sous le radical. C'est cela qui augmente la difficulté de sa résolution.

De ce fait, nous allons étudier dans cette section le comportement asymptotique de la probabilité optimale du gain de l'IDS p^* en fonction de n et g.

Nous allons montrer que $p^*(n, g)$ tend vers 1 lorsque g devient très grand et que $p^*(n, g)$ tend vers 0 lorque n devient très grand.

Pour ce faire nous allons procéder comme suit :

– Pour montrer que quelque soit la valeur de n, $p^*(n, g)$ tend vers 1 lorsque g tend vers l'infini, nous allons montrer que le contraire n'est pas vrai.

  Supposons pour n fixé que $p^*(n, g) < p_0 < 1$ pour des valeurs de g arbitrairement grandes. La valeur du gain net espéré de l'IDS obtenue avec p^* est inférieure ou égale à p^*g, donc $V(p^*, q^*) \leq p^*g \leq p_0 g$.

  D'un autre côté, nous pouvons dire que la valeur du gain net espéré de l'IDS obtenue avec $p = 1$ est supérieure ou égale à $g - n$ ou encore $V(1, q(1)) \geq g - n$. On a donc pour des valeurs de g arbitrairement grandes l'inéquation $p_0 g \geq g - n$, ce qui veut dire que n est supérieur à $(1 - p_0)g$.

  On remarque bien que ce résultat est impossible pour des valeurs de g suffisament grandes.

En conclusion, nous dirons que

$$\forall n \lim_{g \to \infty} p^*(n, g) = 1$$

– Pour montrer que quelque soit la valeur de g, $p^*(n, g)$ tend vers 0 lorsque n tend vers l'infini, nous allons montrer que le contraire n'est pas vrai.

Supposons pour g fixé que la double inégalité $p^*(n, g) > p_1 > 0$ est vraie pour des valeurs de n arbitrairement grandes.

Pour $q = 0$, le jeu doit durer n coups, donc $V(p^*, 0) \leq g - p^* n < g - p_1 n$. Pour n suffisament grand, cette valeur est négative. D'un autre côté nous savons que la valeur du gain net espéré de l'IDS obtenue avec $p = 0$ est supérieure ou égale à 0 pour chaque g ou encore $V(0, q) \geq 0$. Donc $V(p^*, 0)$ ne peut pas être négatif.

Cette contradiction implique :

$$\forall g \lim_{n \to \infty} p^*(n, g) = 0$$

4.6 Version à probabilités dynamiques du jeu $J(n, 1, 1, g)$

Dans cette section nous allons étudier le jeu $J(n, 1, 1, g)$ dans le cas où l'IDS peut changer la probabilité d'activité p à chaque coup. Cette version sera dénotée $\widetilde{J}(n, 1, 1, g)$. Pour ce faire, nous allons déterminer les valeurs des probabilités qui assurent le meilleur gain à l'IDS en pire cas dans les différents coups i. Pour résoudre ce cas particulier nous allons utiliser le résultat obtenu plus haut dans le paragraphe 4.3. Dans ce dernier, nous avons étudié le cas où la taille du jeu, la taille de l'attaque et le nombre de paquets que l'IDS doit intercepter pour détecter l'attaque sont egaux à 1. Nous avons montré que la probabilité de l'activité de l'IDS qui lui assure le meilleur gain en pire cas pour des valeurs de $g > 1$ est $p^* = 1$. Nous avons aussi trouvé que dans ce cas le meilleur gain net espéré de l'IDS est $V(p^*, q^*) = g - 1$. Nous allons utiliser ce résultat pour montrer par induction sur n la meilleure probabilité de l'activité de l'IDS dans chaque coup.

Définissons le coup n comme celui au moment duquel le jeu va durer encore n rounds. Donc les numéros des coups décroient de n à 1.

La figure (4.5) représente les différentes possibilités de jeux de l'IDS et de l'intrus du coup n au coup $(n - 1)$. Chaque branche représente l'intervention des deux joueurs. La branche

(2), par exemple, signifie que l'IDS est actif et l'intrus est passif.

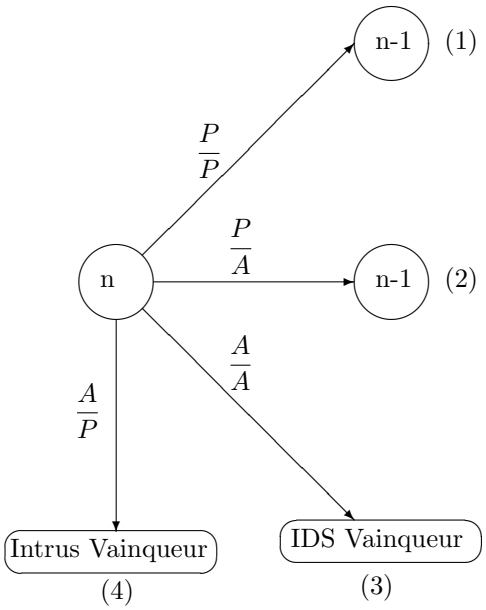

FIGURE 4.5 – Les quatre différents coups conjoints possibles de l'IDS et de l'intrus

On remarque que la probabilité de continuer le jeu(branches (1) et (2)) est égale à $1 - q$, puisque au coup n l'intrus doit alors être passif.

Puisque l'IDS ne peut être actif qu'une seule fois en passant du coup n au coup $n - 1$, le coût espéré du coup n est p.

Supposons que le gain net espéré de l'IDS au $n^{ième}$ coup est noté V_n. Le calcul de V_n ce fait ainsi :

La partie de V_n correspondant aux branches (1) et (2) après le coup n est $(1 - q)V_{n-1}$. Le gain brut correspondant à la branche (3) est pqg et à la branche (4) est 0. En ajoutant ces valeurs et soustrayant le coût espéré p dans le coup n on obtient :

$$V_n = (1 - q)V_{n-1} + pqg - p$$

$$V_n = V_{n-1} - p + q(pg - V_{n-1})$$

Nous considérons d'abord le cas $g > 1$. Pour ce faire nous allons traiter les deux cas suivant :
1^{er}cas : Si $(pg - V_{n-1}) \geq 0$ $\Rightarrow$p$\geq$ $\dfrac{V_{n-1}}{g}$:
Ici la valeur V_n est minimale pour $q^* = 0$. Cela nous donne $V_n = V_{n-1} - p$, qui est maximisé lorsque p est minimal, c'est à dire $p^* = \dfrac{V_{n-1}}{g}$. Cela nous donne la valeur du gain net espéré $V_n = \dfrac{V_{n-1}}{g}(g-1)$.
$2^{ème}$cas : Si $(pg - V_{n-1}) \leq 0$ $\Rightarrow$p$\leq$ $\dfrac{V_{n-1}}{g}$:
Là par contre V_n est minimale pour $q^* = 1$. Cela nous donne $V_n = p(g-1)$, qui est maximisé lorsque p est maximale, c'est à dire $p^* = \dfrac{V_{n-1}}{g}$. Cela nous donne la valeur du gain net espéré $V_n = \dfrac{V_{n-1}}{g}(g-1)$.
Notons la probabilité p^* au $n^{ième}$ coup par p_n. Comme nous avons montré avant on a $V_1 = g - 1$ et $p_1 = 1$.
On a donc $V_n = \dfrac{(g-1)^n}{g^{n-1}}$.
Puisque $p_n = \dfrac{V_{n-1}}{g}$, cela implique que $p_n = \dfrac{(g-1)^{n-1}}{g^{n-2}g}$
ou encore $p_n = \left(\dfrac{g-1}{g}\right)^{n-1}$.
En conclusion nous dirons que si $g > 1$ et dans le cas où l'IDS peut choisir une probabilité à chaque coup, pour le jeu $\widetilde{J}(n, 1, 1, g)$, la probabilité d'activité de l'IDS au coup n est

$$p_n = \left(\frac{g-1}{g}\right)^{n-1}$$

et la valeur du gain net espéré est

$$V_n = \frac{(g-1)^n}{g^{n-1}}$$

Il reste à considérer le cas $0 < g \leq 1$.
Pour $n = 1$, comme on l'a vu dans le paragraphe 4.3, la valeur du gain net espéré de l'IDS est $V_1 = 0$. Dans ce cas nous avons $q^*(p) = 1$ et $p^* = 0$.
Nous allons prouver par induction que $V_n = 0$.
Pour $n > 1$ on a $V_n = (1-q)V_{n-1} + pqg - p$.
Par hypothèse inductive ça implique $V_n = pqg - p$.
Cette formule est minimisée, pour chaque p, lorsque $q = 0$ et sa valeur devient $-p$, ce qui est minimisé lorsque $p = 0$. On obteint $V_n = 0$.
Il s'ensuit que lorsque $g \leq 1$, la meilleure stratégie de l'IDS est de ne jamais être actif.

4.7 Résolution du jeu $J(2, 2, 1, g)$

Dans cette section nous allons déterminer la valeur p^* de la probabilité d'activité de l'IDS qui lui assure le meilleur gain en pire cas pour un jeu où :

– la taille n est égale à 2,

– la taille a de l'attaque est de 2,

– une intrusion est détectée si l'IDS intercepte un paquet ($b = 1$),

– le gain g de l'IDS lorsqu'il intercepte une attaque est supérieur ou égal à zéro

4.7.1 Calcul de la formule du gain net espéré $V(p, q) = G(p, q) - C(p, q)$

Par les équations (4.1) et (4.16), nous avons :

$$G(p,q) = \left(1 - (1-p)^2 q^2 \sum_{k=2}^{2} \binom{k-1}{1} (1-q)^{k-2}\right) g \tag{4.23}$$

et

$$C(p,q) = \sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,A_k(IDS)}^{A_k(Int)} + \sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,P_k(IDS)}^{A_k(Int)} + \sum_{i=1}^{2} C_{2,i} \tag{4.24}$$

D'un côté le développement de l'équation (4.23) nous donne

$G(p,q) = \left(1 - (1-p)^2 q^2 (\binom{1}{1}(1-q)^0)\right) g$

$= \left(1 - (1-p)^2 q^2\right) g$

En d'autres termes

$$G(p,q) = g - g(1-p)^2 q^2 \tag{4.25}$$

De l'autre côté le développement de l'équation (4.24) nous donne

(a) $\sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,A_k(IDS)}^{A_k(Int)} = \binom{0}{0} p(1-p)^0 q(1-q)^0 (\binom{0}{0} q^0 (1-q)^0 + \binom{1}{0} q^0 (1-q)^0)$

$= pq(1+0)$

En d'autres termes

$$\sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,A_k(IDS)}^{A_k(Int)} = pq,$$

(b) $\sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,P_k(IDS)}^{A_k(Int)} = \binom{0}{1} p(1-p)^0 \binom{-1}{0} q(1-q)^{-1}$

Puisque $\binom{0}{1} = 0$ cela implique

$$\sum_{k=1}^{1}\sum_{i=1}^{k} C_{k,i,P_k(IDS)}^{A_k(Int)} = 0,$$

(c) $\sum_{i=1}^{2} C_{2,i} = \binom{1}{0}p(1-p)(1-q)^0(\binom{1}{0}q^0(1-q) + \binom{1}{0}q(1-q)^0)+$
$2\binom{1}{1}p^2(1-p)^0(1-q)(\binom{0}{0}q^0(1-q)^0 + \binom{0}{1}q^0(1-q)^{-1})+$
$\binom{1}{1}p(1-p)(1-q)(\binom{0}{0}q^0(1-q)^0 + \binom{0}{1}q(1-q)^{-1})+$
$2\binom{1}{2}p^2(1-p)^0(1-q)^2(\binom{0}{0}q^0(1-q)^{-1} + \binom{0}{1}q(1-q)^{-2})$

Puisque $\binom{1}{2} = \binom{0}{1} = 0$on aura :
$\sum_{i=1}^{2} C_{2,i} = \binom{1}{0}p(1-p)(1-q)^0(\binom{1}{0}q^0(1-q) + \binom{1}{0}q(1-q)^0)+$
$2\binom{1}{1}p^2(1-p)^0(1-q)(\binom{0}{0}q^0(1-q)^0) + \binom{1}{1}p(1-p)(1-q)(\binom{0}{0}q^0(1-q)^0)$

$\sum_{i=1}^{2} C_{2,i} = p(1-p)(1-q) + p(1-p)q + 2p^2(1-q) + p(1-p)(1-q)$

$\sum_{i=1}^{2} C_{2,i} = pq + 2p - p^2q$

À partir de là, nous dirons que le coût espéré de l'IDS pour le jeu $j(2,2,1,g)$ est :

$$C(p,q) = 2p - p^2q \tag{4.26}$$

Les équations (4.25), (4.26) et la fonction du gain net espéré définie au paragraphe 3.2 impliquent que

$$V(p,q) = -g(1-p)^2q^2 + p^2q - 2p + g \tag{4.27}$$

Nous allons donc dans un premier temps chercher la valeur $q^*(p)$ qui minimise la fonction de gain net espéré de l'IDS pour chaque p. Par la suite, nous allons calculer la valeur p^* de la probabilité de l'activité de l'IDS qui maximise $V(p, q^*(p))$.

4.7.2 Probabilité de l'activité de l'intrus qui minimise le gain net espéré de l'IDS : $q^*(p)$

Dans ce cas la dérivée $\dfrac{\partial V}{\partial q} = -2g(1-p)^2q + p^2$ s'annule pour $q = \dfrac{p^2}{2g(1-p)^2}$.
D'un autre côté, la dérivée seconde $\dfrac{\partial^2 V}{\partial^2 q} = -2g(1-p)^2$ est toujours négative. Ce qui

implique que $V(p,q)$ atteint son maximum pour $q = \dfrac{p^2}{2g(1-p)^2}$. Cela veut dire que le minimum de $V(p,q)$ est atteint soit pour $q = 0$, soit pour $q = 1$. Pour en décider laquelle des valeurs de q minimise la fonction du gain net de l'IDS, nous allons étudier le signe de $D(p) = V(p,1) - V(p,0)$. Pour ce faire nous allons chercher les racines de $D(p)$.
Nous avons $D(p) = (1-g)p2 + 2gp - g$ qui s'annule pour

$$p_1 = \frac{g - \sqrt{g}}{g-1} \text{ et } p_2 = \frac{g + \sqrt{g}}{g-1}.$$

Il y a trois remarques qui découlent de ces résultats. La première nous dit qu'on doit tenir compte de p_1 puisqu'il est compris entre 0 et 1 ($0 \leq p_1 \leq 1$). La deuxième remarque exclut p_2 de notre calcul puisque $p_2 \geq 1$. En fin la troisième remarque nous signifie que le signe de $D(p)$ dépend des valeurs de g et de p.

- Dans le cas où $g > 1$:
  - Si p est compris entre 0 et p_1, $D(p)$ est inférieur à 0. Ce qui veut dire que $V(p,1)$ est inférieur à $V(p,0)$.

    On a donc la valeur de la probabilité d'activité de l'intrus qui minimise le gain net espéré de l'IDS est $q^*(p) = 1$.
  - Si p est compris entre p_1 et 1, $D(p)$ est supérieur à 0. Ce qui veut dire que $V(p,1)$ est supérieur à $V(p,0)$.

    On a donc la valeur de la probabilité d'activité de l'intrus qui minimise le gain net espéré de l'IDS est $q^*(p) = 0$.
- Dans le cas où $g < 1$:
  - Si p est compris entre 0 et p_1, $D(p)$ est supérieur à 0. Ce qui veut dire que $V(p,1)$ est supérieur à $V(p,0)$.

    On a donc la valeur de la probabilité d'activité de l'intrus qui minimise le gain net espéré de l'IDS est $q^*(p) = 0$.
  - Si p est compris entre p_1 et 1, $D(p)$ est inférieur à 0. Ce qui veut dire que $V(p,1)$ est inférieur à $V(p,0)$.

    On a donc la valeur de la probabilité d'activité de l'intrus qui minimise le gain net espéré de l'IDS est $q^*(p) = 1$.
- Dans le cas où $g = 1$, on a $V(p,0) = 1 - 2p$ et $V(p,1) = 0$. La fonction $V(p,0)$ est une fonction décroissante, elle décroit de $V(0,0) = 1$ jusqu'à $V(1,0) = -1$ et elle s'annule pour $p = \dfrac{1}{2}$.

On remarque que le maximum de $V(p,1)$ est toujours inférieur au maximum de $V(p,0)$. Ce qui veut dire que la valeur de la probabilité d'activité de l'intrus qui minimise le gain net espéré de l'IDS est $q^*(p) = 1$

4.7.3 Probabilité de l'activité de l'IDS qui maximise la gain net espéré de l'IDS : p^*

Pour calculer p^* trois cas sont à prendre en considération :

- Dans le cas où $g > 1$, on remarque de ce qui précède que :
  - Pour p compris entre 0 et p_1, la fonction $V(p,1) = (1-g)p^2 - 2(1-g)p$ est croissante, elle croit de $V(0,1) = 0$ jusqu'à $V(p_1,1) = \dfrac{g^2 - 3g + 2\sqrt{g}}{g-1}$.
  - Pour p compris entre p_1 et 1, la fonction $V(p,0) = g - 2p$ est décroissante, elle décroit de $V(p_1,0) = \dfrac{g^2 - 3g + 2\sqrt{g}}{g-1}$ jusqu'à $V(1,0) = g - 2$.
  - La fonction $V(p,q^*(p))$ qui croit de $V(0,1)$ jusqu'à $V(p_1,q^*(p_1))$ et en suite elle décroit jusqu'à $V(1,0)$. Ce qui implique que $V(p,q^*(p))$ atteint son maximum pour $p = p_1$.

  En conclusion nous dirons que pour $g > 1$, la valeur de la probabilité de l'activité de l'IDS qui maximise la fonction du gain net espéré de l'IDS est :

$$p^* = \frac{g - \sqrt{g}}{g-1}$$

Le graphique suivant montre bien la variation de $V(p,q^*(p))$ en fonction de p

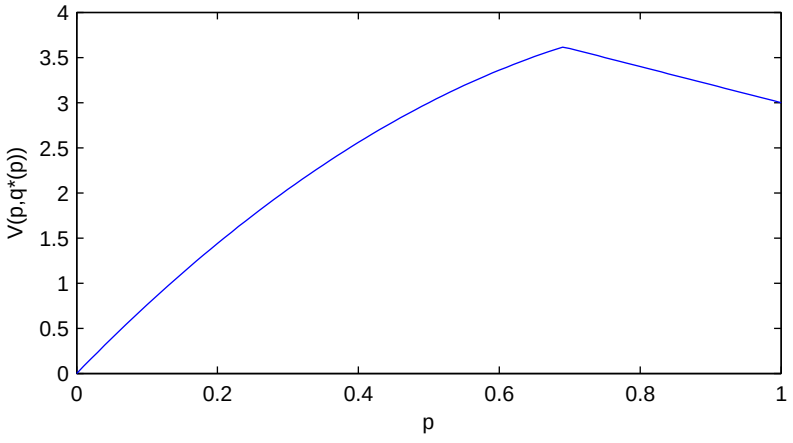

FIGURE 4.6 – Graphe de $V(p,q^*(p))$ pour $g = 5$

– Dans le cas où $g < 1$, On remarque de ce qui précède que :
  – Pour p compris entre 0 et p_1, la fonction $V(p,0) = g - 2p$ est décroissante, elle décroit de $V(0,0) = g$ jusqu'à $V(p_1,0) = \dfrac{g^2 - 3g + 2\sqrt{g}}{g-1}$.
  – Pour p compris entre p_1 et 1, la fonction $V(p,1) = (1-g)p^2 - 2(1-g)p$ est décroissante, elle décroit de $V(p_1,0) = \dfrac{g^2 - 3g + 2\sqrt{g}}{g-1}$ jusqu'à $V(1,1) = g - 1$.
  – Les deux points précédents signifient que le maximum de la valeur du gain net espéré de l'IDS est $V(0,0) = g$.

  En conclusion nous dirons que puisque pour $g < 1$, la valeur de la probabilité de l'IDS qui maximise son gain net espéré en pire cas est :

$$p^* = 0$$

  Le graphique représentant la variation de $V(p, q^*(q))$ confirme que pour $g < 1$ le meilleur gain pour l'IDS se réalise pour $p^* = 0$

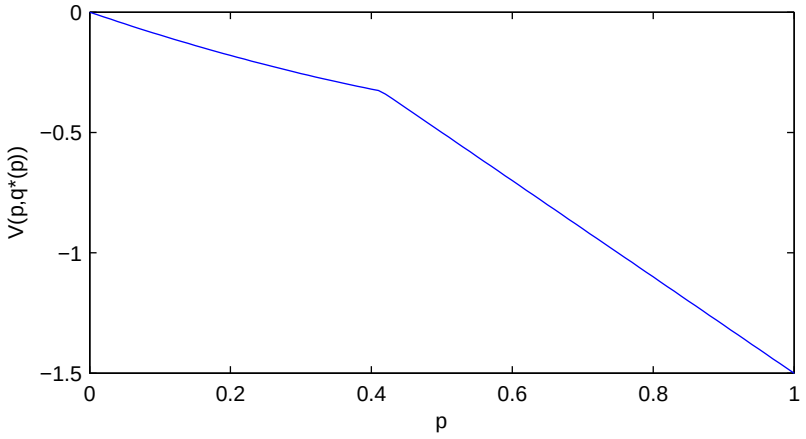

FIGURE 4.7 – Graphe de $V(p, q^*(p))$ pour $g = \dfrac{1}{2}$

– Dans le cas où $g = 1$, puisque $V(p,1) = 0$ pour $p \leq \dfrac{1}{2}$ et que $V(p,0)$ est négatif pour $p > \dfrac{1}{2}$, il suffit de prendre la valeur de p inférieure ou égale à $\dfrac{1}{2}$ pour maximiser le gain net espéré de l'IDS. En conclusion nous dirons que pour $g = 1$, la valeur de la probabilité de l'activité de l'IDS qui maximise la fonction du gain net espéré de l'IDS est :

$$p^* \leq \frac{1}{2}$$

Dans ce cas aussi, le graphe de $V(p, q^*(p))$ montre bien que le meilleur gain de l'IDS est atteint pour $p^* \leq \frac{1}{2}$

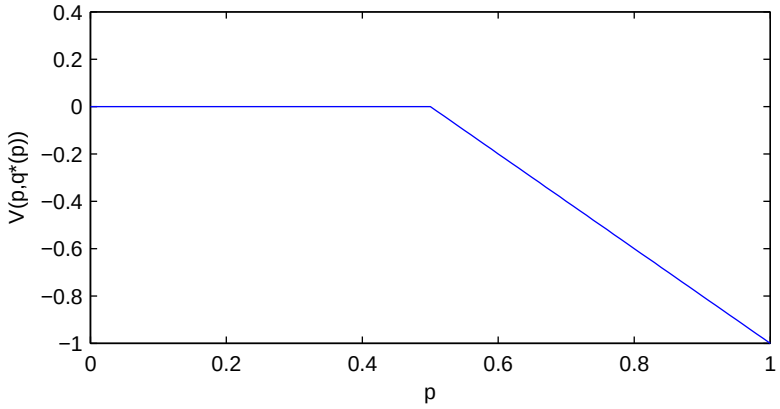

FIGURE 4.8 – Graphe de $V(p, q^*(p))$ pour $g = 1$

Chapitre 5

Résolution du jeu $J(n, n, n, g)$

5.1 Introduction

Dans ce chapitre nous allons étudier le jeu qui dure n coups où l'attaque tient sur n paquets et pour gagner la partie, l'IDS doit intercepter tous les paquets. Ceci dit, l'intrus doit aussi être actif durant toute la partie pour faire passer les n paquets de l'attaque.

5.2 Formule générale calculant $V(p, q)$

Dans le chapitre 3, nous avons vu que le gain net espéré de l'IDS $V(p, q)$ est obtenu par la différence entre le gain brut espéré $G(p, q)$ et le coût espéré de l'IDS $C(p, q)$. Nous avons aussi défini le gain brut espéré et le coût espéré de l'IDS.

5.2.1 Calcul du gain brut espéré $G(p, q)$

Comme nous l'avons défini plus haut, le gain brut espéré est obtenu par le produit de la probabilité du gain de l'IDS p_g et la valeur g du gain de l'IDS lorsqu'il intercepte une attaque.
Nous remarquons bien que dans le cas que nous traitons dans ce chapitre, l'IDS perd quand il n'est pas actif durant tous les n coups que dure le jeu et que l'intrus quand à lui il est tout le temps actif. Cette perte se réalise avec un probabilité de $(1 - p^n)q^n$. Ceci dit, la probabilité de gain de l'IDS est :

$$p_g = (1 - (1 - p^n)q^n)$$

Nous avons donc, montré que dans un jeu qui dure n coups où l'attaque se fait avec n paquets et que l'IDS doit intercepter tous les paquets pour débusquer une intrusion, le

gain brut espéré de l'IDS est donné par :

$$G(p,q) = (1-(1-p^n)q^n)g$$

5.2.2 Calcul du coût espéré de l'IDS $C(p,q)$

Dans ce cas, déterminer le coût espéré de l'IDS revient à déterminer l'espérance mathématique du nombre de succès dans une série de Bernoulli. Cela est dû au fait que d'un côté, le jeu doit se terminer au $n^{ième}$ coup. De l'autre côté à chaque coup l'IDS doit être soit actif avec une probabilité p soit passif avec une probabilié $(1-p)$. Nous pouvons donc dire que l'IDS est en situation de sucès avec un probabilité p et en situation contraire avec une probabilité $(1-p)$. Nous pouvons associer une variable aléatoire de Bernoulli X_i à chaque coup i. Cette dernière sera égale à A lorsque l'IDS est actif et elle sera égale à P lorsque l'IDS est passif. La probabilité d'activité de l'IDS au $i^{ème}$ sera exprimée par :

$$P[X_i = A] = p$$

et

$$P[X_i = P] = 1-p$$

Pour une variable aléatoire de Bernoulli X_i, l'espérance mathématique est $E(X_i) = p$. Le nombre d'activités de l'IDS est représenté par la variable $X = \sum_{i=1}^{n} X_i$.
On aura l'espérance mathématique $E(X) = \sum_{i=1}^{n} E(X_i)$ qui nous donnera $E(X) = np$.
En conclusion nous dirons que pour un jeu qui dure n coups où l'attaque se fait avec n paquets et que l'IDS doit intercepter tous les paquets pour débusquer une intrusion, le coût espéré de l'IDS est :

$$C(p,q) = np$$

5.2.3 Gain net espéré

Pour un jeu qui dure n coups où l'attaque se fait avec n paquets et que l'IDS doit intercepter tous les paquets pour gagner la partie, le gain net espéré de l'IDS est donnée par :

$$V(p,q) = (1-(1-p^n)q^n)g - np$$

5.3 Résolution du jeu

Dans cette section nous allons chercher la valeur p^* de la probabilité d'activité de l'IDS qui lui assure le meilleur gain en pire cas. Pour ce faire nous allons d'abord déterminer le pire cas pour l'IDS en calculant, pour toutes les valeurs possibles de p, la valeur de la probabilité d'activité de l'intrus $q^*(p)$ qui minimise la valeur du gain net espéré de l'IDS. En nous basant sur cette valeur $q^*(p)$, nous allons déterminer p^*.

5.3.1 Calcul de $q^*(p)$

En examinant la formule du gain net espéreé de l'IDS, $V(p,q) = -g(1-p^n)q^n - np + g$, calculée plus haut, nous remarquons que la valeur de la probabilité q qui la minimise est $q = 1$ quelque soit la valeur de p. Cela veut dire que :

$$q^*(p) = 1$$

5.3.2 Calcul de p^*

Pour calculer la valeur de la probabilité de l'IDS p^* qui maximise son gain en pire cas, nous allons remplacer la variable q dans la la formule $V(p,q)$ par 1. Nous allons, ensuite, chercher la valeur de p qui maximise $V(p,1)$.
Nous avons $V(p,1) = gp^n - np$. D'un côté la dérivée de $V(p,1)$ donnée par
$V'(p,1) = ngp^{n-1} - n$ s'annule pour $p = \frac{1}{\sqrt[n-1]{g}}$. De l'autre côté la dérivée seconde de $V(p,1)$ donnée par $V''(p,1) = n(n-1)gp^{n-2}$ est toujours positive, ce qui veut dire que la valeur de p qui annule la dérivée de $V(p,1)$ minimise le gain net espéré de l'IDS.
Il est donc évident que la valeur de la probabilité p qui maximise le gain net espéré de l'IDS va être soit $p = 0$ soit $p = 1$. En remplaçant p par ces deux valeurs dans la formule $V(p,1)$ nous aurons :

$$V(0,1) = 0$$

et

$$V(1,1) = g - n$$

Donc pour $g > n$ on a $p^* = 1$ et pour $g \leq n$, on a $p^* = 0$.

5.3.3 Conclusion

Pour un jeu qui dure n coups où l'attaque tient sur n paquets et pour gagner la partie, l'IDS doit intercepter tous les paquets, la meilleure stratégie pour ce dernier dépend de la valeur du gain g. Dans le cas où g est supérieur à n, l'IDS va tout le temps être actif, $p^* = 1$. Dans les autres cas, il va être passif, $p^* = 0$.

Chapitre 6

Conclusion

Dans ce mémoire, nous avons étudié la problématique de modélisation des IDS en utilisant la théorie des jeux. En particulier, nous nous intéressions au calcul de stratégies optimales dans un modèle de jeux probabilistes. Pour ce faire, nous avons commencé par présenter une revue de littérature pour les systèmes de détection d'intrusions existants. Dans un premier temps, nous avons défini la détection d'intrusion, le fonctionnement des IDS et leurs catégories. Par la suite, nous avons passé en revue l'évolution de la technologie en présentant les solutions les plus pertinentes. Par ailleurs, nous nous sommes penchés sur les contributions qui traitent la modélisation des IDS en utilisant la théorie des jeux. Sur ce sujet, nous avons remarqué que la plupart des solutions existantes traitent le problème en se basant sur l'architecture réseau, le trafic entre les nœuds, les sessions réseau et bien sûr sur le comportement des deux joueurs, l'IDS et l'intrus.

À partir de ce constat nous avons orienté nos recherches vers la modélisation du problème en un jeu entre l'IDS et l'intrus en tenant compte uniquement de l'activité des joueurs. Notre jeu se base sur un modèle probabiliste. Ce choix est motivé par le souci de trouver un moyen qui permettra à l'IDS de choisir une stratégie de jeu qui lui garantit les meilleurs gains. L'IDS choisit cette stratégie en calculant la fréquence de ses activités qui permet d'avoir le meilleur gain net en pire cas. Dans les résultats que nous avons présentés plus haut, nous sommes arrivés à démontrer que notre modèle permet à l'IDS de choisir dans plusieurs cas la stratégie qui lui offre le meilleur gain en optimisant le coût de ses activités.

Nous prévoyons que notre approche qui combine la détection probabiliste d'intrusion et la théorie des jeux contribuera à améliorer le taux de détection d'intrusion tout en optimisant son coût. Ceci dit, le problème que nous avons taité reste ouvert. En effet, il

serait intéressant de traiter le jeu $J(n, a, b, g)$ où l'IDS doit intercepter plus qu'un paquet pour gagner.

Bibliographie

[1] T. Alpcan et T. Basar. *A game theoretic approach to decision and analysis in network intrusion detection*. In Proc. of the 42nd IEEE Conference on Decision and Control, pages : 2595 - 2600, Maui, Hawaii, décembre 2003.

[2] T. Alpcan et T. Basar. *A game theoretic analysis of intrusion detection in access control systems*. In Proc. of the 43rd IEEE Conference on Decision and Control, page(s) : 1568-1573, Paradise Island, Bahamas, décembre 2004.

[3] T. Alpcan et T. Basar. *An intrusion detection game with limited observations*. In 12th Int. Symp. on Dynamic Games and Applications, Sophia Antipolis, France, july 2006.

[4] J.P. Anderson. *Computer Security Threat Monitoring and Surveillance*. Technical Report, J.P. Anderson Company, Fort Washington, Pennsylvania, 1980.

[5] R. Bace, P. Mell. *Intrusion detection systems*. National Institute of Standards and Technology, special publication on intrusion detection systems, Gaithersburg, Maryland, 2001.

[6] N. Berlin et al. *Théorie des jeux : Introduction à la théorie des jeux répétés*, Paris, Édition de l'École Polytechnique, janvier 2007.

[7] M. Couture. *Détection des variations d'attaques à l'aide d'une logique temporelle*. Mémoire de maîtrise en informatique de l'université de Laval, juin 2005.

[8] F. Cuppens. *Managing Alerts in a Multi-Intrusion Detection Environment*. In 17th Annual Computer Security Applications Conference, New-Orleans, décembre 2001.

[9] D. Curry et H. Debar. *Intrusion Detection Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition*. Intrusion Detection Working Group. Merrill Lynch et France Telecom, 2002

[10] H. Debar. *Corrélation d'alertes et cartographie de sites*. Projet RNTL DICO France Télécom 22 mai 2002. Publié sur http ://www.lsv.ens-cachan.fr/ goubault/DICO/DICO/SP5_1_draft_V01.pdf, 2002

[11] D.E. Denning *An Intrusion Detection Model*, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL :SE-13, issue : 2 ; pages : 222-232 , 1987.

[12] T. Evangelista. *Les systèmes de détection d'intrusions informatiques*. Paris : Dunod, 2004.

[13] L. Hamza, 2005. *Génération automatique de scénario d'attaques pour les systèmes de détection d'intrusions*. Mémoire de magister, Béjaia, Université A. Mira de Béjaia.

[14] Kasperky Lab. *Eviter la majorité des infections virales tout en protégeant votre ordinateur*. Publié sur le site Internet : http ://kb.kaspersky.fr/index.php ?article=583onglet=4

[15] G. Lehmann. *Généralités sur les systèmes de détection d'intrusions*. Publié le 13 Avril 2003 sur le site Internet : http ://lehmann.free.fr/RapportMain/node10.html

[16] P.L Lespérance. *Détection d'intrusions et analyse passive du réseaux*. Mémoire de maîtrise en informatique de l'université de Laval, août 2005.

[17] N. Marchang et R. Tripathi. *A Game Theoretical Approach for Efficient Deployment of Intrusion Detection System in Mobile Ad Hoc Networks*. In 15th International Conference on Advanced Computing and Communications, pages :460-464, Guwahati, India, décembre 2007.

[18] M. Mehrandish et al.. *A Game Theoretic Model to Handle Network Intrusions over Multiple Packets*. In 2006 IEEE International Conference on Communications vol. 5, page(s) :2189 - 2194 , Istanbul, Turky, juin 2006.

[19] M. Mehrandish et al.. *A Game Theoretic Approach to Detect Network Intrusions : The Cooperative Intruders Scenario*. In IEEE Global Telecommunications Conference, pages 1-5, San Francisco, décembre 2006.

[20] L. Mé, 1997. *Un complément de l'approche formelle : la détection d'intrusions*. Journée CIDR97, Rennes, octobre 1997. Publié sur http ://www.rennes.supelec.fr/rennes/si/equipe/lme/

[21] B. Morin. *Corrélation d'alertes issues d'outils de détection d'intrusions avec prise en compte d'informations sur le système surveillé*. Thèse de doctorat en informatique de l'institut nationnal des sciences appliqueés de Rennes, février 2004.

[22] K. Murali et T.V. Lakshman, 2003. *A Detecting network intrusions via sampling : a game theoretic approach*. In the twenty-Second Annual Joint Conference of the IEEE Computer and Communications, vol.3, pages :1880 - 1889, San Francisco, avril 2003

[23] D. K. Shneider. *Modélisation de la démarche du décideur politique dans la perspective de l'intelligence artificielle*. Thèse de doctorat ès sciences économiques et sociales, mention science politique. Université de Genève Faculté des Sciences économiques et sociales Département de science politique, septembre 1994.

[24] J. Zimmermann et L. Mé. . *Les systèmes de détection d'intrusions : principes algorithmiques*. Publié dans le magazine "Multi-System & Internet Security Cookbook", numéro 3, pages : 24-30, juin 2002.

[25] J. Zimmermann. *Détection d'intrusions paramétrée par la politique par le contrôle de flux de références*. Thèse de doctorat de l'université de Rennes, décembre 2003.

[26] The National Information Assurance Partenership. *Cisco Intrusion Detection System Sensor Appliance IDS-4215 series Version 4.1(3) Security Target.* . Manual of Cisco, 2004.