الجمهورية الجزَائِرية الديمقرَاطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزَارة التعليم العَالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جَامعة عمَار ثليجي بِالأَغوَاط
UNIVERSITE AMAR TELIDJI LAGHOUAT

FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

PROJET DE FIN D'ETUDE (MASTER)

Filière **:** Informatique

Option **:** Réseaux, Systèmes et Applications Réparties

Thème

CONCEPTION D'UN SYSTÈME D'AIDE À LA DÉCISION BASÉ SUR LES HONEYPOTS DISTRIBUÉS

Présenté Par

Nour Elhouda OUDENANI

Devant le jury composé de :

Dr. Noureddine CHAIB	Président	MCB, Université de Laghouat
Dr. Younes GUELLOMA	Examinateur	MCB, Université de Laghouat
Dr. Mustapha BOUAKKAZ	Encadrant	MCA, Université de Laghouat
Dr. Taha Elamine HADJADJ	Co-Encadrant	Ecole Superieure de Communication Tunis, Tunisie

Année Universitaire 2018/2019


بسم الله الرحمن الرحيم

Dédicaces

Je dédie ce travail

A mes très chers parents

A mon très cher mari

A mes soeurs et frères

A toute ma famille

Remerciement

En premier lieu, nous remercions ALLAH qui nous a permis d'arriver jusque-là,
Grand merci à mes parent ,pour l'orientation, la confiance, la patience et le soutien qui ont constitué un apport considérable

Nous tiendrons à exprimer nos sincères remerciements à notre encadrant Mr. Mustapha Bouakkaz et notre Co-encadrant Mr. Taha Hadjadj. Pour leur aide sans limites et leurs précieux conseils sans lesquels ce travail n'aurait pas vu le jour.

Nos vifs remerciements vont également aux membres du jury
pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et pour la grande attention qu'ils ont bien voulu porter.

Un grand merci à ma famille pour son soutien tout au long de ma scolarité, jusqu'à ce mémoire.
Je présente mes sincères remerciements à tous les enseignants et responsables du département Math et Informatique de l'université de Laghouat

À tous ceux qui m'ont aidé, que ce soit par une grande collaboration ou par une tendre parole.

ملخص

مع تطور التكنولوجيا واستخداماتها المتعددة في حياتنا ، أصبح أمن المعلومات من أهم القضايا التي تهم الفنيين والمديرين وضمان سلامة بياناتهم وسريتها. كما أصبح تطويرها الطموح الأكثر أهمية.

إن استخدام الحماية وحدها الآن ، مثل الحماية من الفيروسات أو استعمال جدار الحماية لا يكفي يمكن استخدام حلول أخرى مثل استخدام قدر العسل أو مصائد مخترقي الشبكات ، والذي يعد بمثابة مصيدة للمتسللين ، للكشف عن الهجمات غير المدرجة في القائمة لإنشاء هيكل لمنع الاختراق للتوزيع و التعاون والقرب من متطلبات نظام الوقاية من الجيل التالي تشمل هذه المتطلبات القدرة على الاستجابة دون تدخل بشري ضد الهجمات. يتضمن مشروعنا مقارنة بين مصائد مخترقي الشبكات المختلفة الموجودة لمساعدة المسؤول على الاختيار وفقًا لاحتياجاته. كما سنطبق أيضًا نظامًا لاتخاذ القرارات الأمنية استنادًا إلى مصائد مخترقي الشبكات الموزعة التي تم جمعها عن طريق العديد من مصائد مخترقي الشبكات الموزعة في الشبكة في خادم مركزي حيث سنقوم بتحليل كل هذه البيانات لتقديم تقرير حول الأمان لمتخذي القرارات.

Résumé

Avec le développement de la technologie et sa multiple utilisation dans notre vie, la sécurité de l'information est devenue l'un des problèmes les plus importants qui préoccupent les techniciens et les gestionnaires et assurer l'intégrité de leurs données et la confidentialité est l'ambition la plus importante.
L'utilisation des moyens de protections seules, tels que les antivirus ou les pare-feu, ne suffit pas ; d'autres solutions telles que l'utilisation du pot miel, qui sert de piège pour les pirates informatiques, peuvent être utilisés pour détecter les attaques non incluses dans la liste afin de créer une structure de prévention de la pénétration destinée à la distribution et à la coopération et à la proximité des exigences du système de prévention de La nouvelle génération. Ces exigences incluent notamment la capacité de réagir sans intervention humaine contre les attaques. Notre projet de mémoire inclut une comparaison entre les différents pots de miel existant pour aider l'administrateur à choisir en fonction de ses besoins.
Nous allons également proposer un nouveau système de décision de sécurité basé sur les pots miel distribués. On récolte les données capturées par plusieurs pots miel dispersés dans le réseau sur un serveur central. Ces données seront ensuite analysées à fin de fournir un rapport sur la sécurité pour les décideurs.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

La popularité croissante d'Internet et des services qu'elle fournit nous a fait naître le besoin de sécurité sur Internet. Ce besoin est en constante évolution et devient critique pour la sécurité de notre monde électronique. La notion de connexion du monde via une infrastructure commune a entraîné l'émergence d'une communauté ciblant par le biais d'activités malveillantes les systèmes informatiques qui composent cette infrastructure, menaçant la sécurité de nos communications et transactions sur Internet [43]. C'est la communauté notoire du «chapeau noir». Divers outils de sécurité ont été développés dans le but de protéger les systèmes informatiques et les utilisateurs contre les attaquants. Les attaquants qui évoluent rapidement et qui proposent de nouveaux outils et stratégies pour prendre les systèmes sous leur contrôle. Néanmoins, le meilleur moyen de se protéger des ennemis est d'en savoir autant que possible sur eux et de se tenir au courant de toutes les méthodes novatrices qu'ils peuvent introduire. L'idée d'attirer les ennemis dans le système, de les impliquer davantage et révéler des informations sur eux-mêmes sans causer de préjudice aux systèmes réels a rapidement émergé et le terme «pot de miel» est devenu une réalité [49] et une nécessité.

Motivation :

Au cours des dix dernières années environ, divers outils et dispositifs de sécurité réseau ont été mis au point dans le but d'atténuer les menaces auxquelles les organisations sont confrontées par le biais d'attaques réseau. Ces outils incluent des pare-feu qui empêchent les attaquants d'atteindre les ressources réseau, les systèmes de détection d'intrusion (IDS) avec la capacité d'identifier les menaces et de réagir en conséquence, entre autres. Cependant, ces dispositifs constituent des outils passifs qui ne permettent pas de détecter de nouveaux types d'attaques de réseau ou de collecter plus d'informations sur les méthodes de l'attaquant. Les pots de miel viennent changer cela, renversant les règles du jeu pour les hackers et les spécialistes de la sécurité des réseaux. Alors que, jusqu'à présent, tous les efforts étaient déployés pour rendre les systèmes informatiques aussi sécurisés que possible, le concept de pots de miel fait apparaître une nouvelle philosophie de sécurité du réseau consistant à laisser délibérément des failles de sécurité et à inviter l'ennemi à attaquer[28]. Chaque paquet destiné aux pots de miel est suspect, car ces systèmes ne servent à rien et ne font partie d'aucun réseau de production. [46].dans le domaine des appliances de sécurité réseau, les pots de miel constituent un outil unique. Hormis leur particularité d'attirer l'attaquant pour qu'il interagisse réellement avec le système, ils ne doivent pas être considérés comme une solution de sécurité à un problème donné, mais plutôt comme un outil flexible pouvant être utilisé de multiples manières différentes. En tant que nouvelle technologie, les pots de miel vont devenir l'une des armes les plus importantes de l'arsenal des experts en sécurité des réseaux.

Problématique :

L'évolution des nouveaux techniques réseaux, communications et traitement des données ont introduit de nouvelles opportunités à notre société et a permis à plus d'objets à se connecter. Mais cette évolution a aussi causé des risques de sécurité plus difficile. La taille des réseaux est en croissance et ils sont composés de plusieurs hôtes qui peuvent être compromis ou utilisés d'une façon non légitime. Pour mieux se protéger, nous avons besoin de :

— Comprendre les attaqueurs (qui ? comment ? et pourquoi ?).

— Monitorer les attaques dans plusieurs endroits répartis dans le réseau.

— Analyser et extraire les informations des attaques pour les prévoir et améliorer la politique de sécurité.

Objectif :

Ce travail est basé sur plusieurs points :

— Comprendre l'utilité et le fonctionnement des honeypots.

— Proposition d'une nouvelle architecture distribuée pour les honeypots.

— Proposition d'une méthode d'extraction et d'analyse des connaissances récoltées depuis le réseau des honeypots, et une réaction défensive comme résultat.

— Implémentation et simulation du système.

Structure du mémoire :

Premièrement, on a commencé par une introduction générale où nous avons montré la problématique et l'objectif de notre travail, puis nous avons introduit quatre chapitres qui sont les suivants :

— Chapitre 1 : **Notions de base.** Dans cette partie nous allons définir tous les notions requise dans notre projet. En commençant par la sécurité des réseaux, les différents équipements et outils et finalement les connaissances au tour des pots de miel.

— chapitre 2 : **État de l'art, étude comparative et travaux connexes.** Dans la deuxième partie de notre mémoire nous présentant une étude approfondie sur l'existant et l'état de l'art avec étude comparative sur les pots de miel et leurs différentes applications dans le domaine de recherche et professionnel.

— Chapitre 3 : **Contribution.** Dans cette partie nous introduisant le côté théorique de notre travail, l'architecture de notre système et ses différents modules et les méthodes de data mining employées.

— Chapitre 4 : **Étude expérimentale.** Finalement dans ce dernier chapitre nous détaillons l'implémentions de chaque module de notre système ainsi que l'algorithme de data mining, nous présentons les différent étapes expérimentaux réalisés et les résultats abouti.

Finalement, on se terminera avec une conclusion générale et quelques perspectives intéressantes concernant ce travail.

Chapitre 1

Notions de base.

Introduction

La sécurité de l'information au sens personnel et institutionnel est devenue une priorité absolue dans le monde moderne numérisé, parallèlement aux nouveaux développements technologiques. De nombreuses méthodes, outils et technologies sont utilisés pour assurer la sécurité de l'information des systèmes informatiques. Celles-ci sont considérées comme des systèmes de chiffrement, d'authentification, de pare-feu et de détection et de prévention des intrusions. De plus, des systèmes de pots de miel sont proposés en tant que structures complémentaires.

1.1 Le réseau

1.1.1 Définition

Le réseau comprend au moins deux ordinateurs associés au partage de ressources (telles que des imprimantes), au partage de fichiers ou à l'autorisation de communications électroniques. Les ordinateurs du réseau peuvent être connectés via des câbles, des lignes téléphoniques, des ondes radio, des satellites ou des faisceaux lumineux infrarouges.

Il existe deux types de réseaux très courants[61] :

Réseau local (LAN) : Le réseau local (LAN) est un réseau limité à une zone relativement petite. Habituellement confiné à une zone géographique telle qu'un laboratoire d'écriture, une école ou un bâtiment.

Réseau étendu (WAN) : Les réseaux étendus (WAN) connectent des réseaux dans des zones géographiques plus vastes, telles que Laghouat, l'Algérie ou le monde. Des câbles personnalisés peuvent être utilisés à travers l'océan ou les connexions en amont du satellite pour connecter ce type de réseau mondial.

1.1.2 Les attaques réseaux

Ce type d'attaque se base principalement sur des failles liées aux protocoles ou à leur implémentation .

Les attaques les plus connues [35] :

IP Spoofing : L'objectif de cette attaque est d'usurper l'adresse IP d'une autre machine dans le but de se faire passer pour cette dernière en truquant les paquets IP.

ARP Spoofing : Dans ce cas, l'objectif est de rediriger le trafic d'une machine vers une autre grâce à cette redirection, une personne mal intentionnée peut se faire passer pour une autre. De plus, l'attaquant peut rediriger les paquets qu'il reçoit vers le véritable destinataire, ainsi l'utilisateur usurpé ne se rendra compte de rien. La finalité est la même que l'IP spoofing mais ici l'attaque se fait au niveau de la couche liaison de données.

DNS Spoofing : Internet étant basé essentiellement sur le système DNS pour la localisation des ressources, le but de cette attaque est de fournir de fausses réponses aux requêtes DNS, c'est-à-dire indiquer une fausse adresse IP pour un nom de ressource (site web, ftp, ...etc.). Ceci fait, les internautes sont renvoyés à leur insu vers des sites compromis (attaque fishing par exemple qui vise les sites des établissements bancaires par exemple). Grâce à cette fausse redirection, l'utilisateur peut envoyer ses identifiants en toute confiance. Les sites des banques sont souvent utilisés pour les attaques de fishing.

Attaques par fragments IP : Le but de cette attaque est de passer outre les protections des équipements de

filtrage IP dans le but de s'infiltrer dans un réseau pour effectuer des attaques ou récupérer des informations confidentielles.

TCP Session Hijacking : Le but de cette attaque est de rediriger un flux TCP afin de pouvoir outrepasser une protection par mot de passe. Le contrôle d'identité s'effectuant uniquement à l'ouverture de la session, un attaquant réussissant cette attaque parvient à prendre possession de la connexion pendant toute la durée de la session.

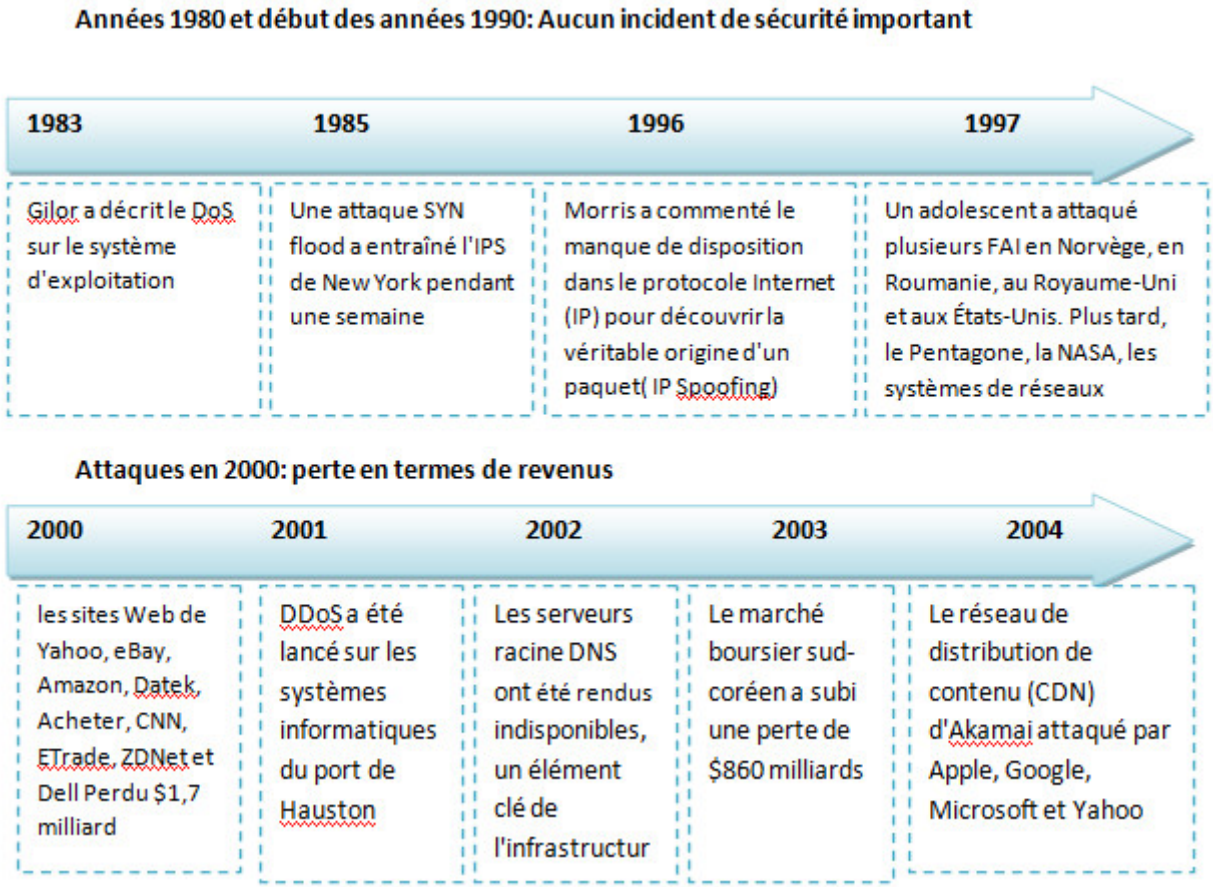

FIGURE 1.1 – L'incidents d'attaque de réseau

[25]

1.2 Pot de miel

Un pot de miel " ou Honeypot " est un programme qui prend l'apparence d'un service attrayant, défini de services, de tout un système d'exploitation ou même de tout un réseau, mais est en réalité un compartiment hermétique construit pour attirer un attaquant.

1.2.1 Modèle générique des pots de miel

Un pot de miel est mis en place sur un réseau dans le seul but d'être attaqué. Il est conçu avec des vulnérabilités délibérées, ce qui est exposé à un réseau public. Aucune valeur de production n'est attribuée à un pot de miel. Les pots de miel ne sont donc pas censés recevoir trafic

légitime. Par conséquent, tout trafic destiné à un pot de miel est très probablement une attaque en cours et peut-être analysé pour révéler vulnérabilités ciblées [25]
Le pot de miel comprend :

— **1 Système de production pot de miel :**
Ce n'est pas un véritable système de production, mais une proie pour les intrus. Cela fournit le miel-fichiers et de fausses ressources système. Automatique les réponses aux actions de l'intrus sont configurées pour montrer le pot de miel comme un véritable système de production.

— **2 Les pare-feu :**
fournissent des journaux indiquant comment un intrus tente de pénétrer dans un pot de miel[40]. Le pare-feu est configuré pour enregistrer tous les paquets envoyés au système pot de miel, car il ne devrait y avoir aucune raison légitime pour que le trafic aille ou vienne du pot de miel.

— **3 Unité de surveillance :**
Il s'agit d'une unité d'évaluation de la menace qui surveille les activités du réseau et / ou du système à la recherche d'activités malveillantes ou de violations des règles et produit des rapports vers une station de gestion. Revoir la commande, de la séquence, de l'horodatage et du type de paquets utilisés par un intrus pour accéder à pot de miel et aux frappes au clavier, aux accès système, aux fichiers modifiés, etc., permet d'identifier les outils, la méthodologie utilisée par les intrus et leurs intentions. (vandalisme, vol de données, recherche de points de lancement à distance, etc.). Un IDS peut faire le travail d'un monitoring unité.

— **4 Unité d'alerte :**
pot de miel devrait être en mesure de générer des alertes par courrier électronique ou par pager afin d'envoyer une notification concernant le trafic à destination ou en provenance du pot de miel à l'administrateur pour lui permettre de contrôler l'activité des intrus PENDANT qu'elle se produise.

— **5 Unité de journalisation :**
cette unité fournit un stockage efficace pour tous les journaux du pare-feu et du système, ainsi que pour le trafic passant entre le pare-feu et le système pot de miel.

1.3 Pots de miel VS Pare-feu et Système de détection d'intrusion (IDS)

Bien que les pots de miel soient utilisés avec le pare-feu et l'IDS, ils ne doivent pas être confondu [44]. Le principe même de chacun est différent. Les systèmes de détection d'intrusion ou les pare-feu ne fournissent qu'une sécurité primitive[25].

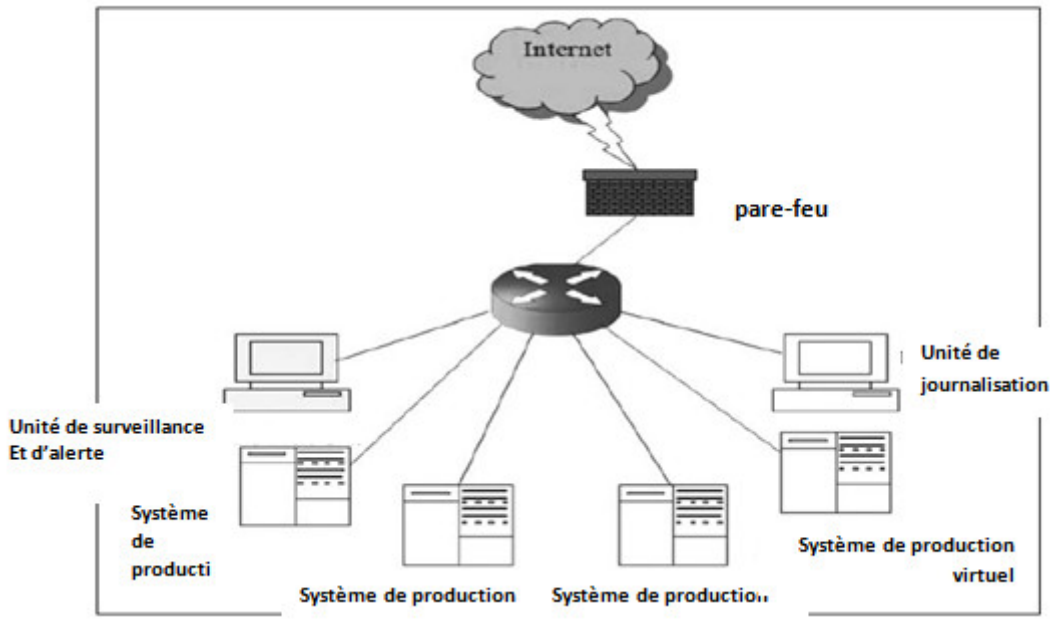

FIGURE 1.2 – modèle générique de pot de miel

[25]

1.3.1 Pare-feu L'utilisation de pare-feu à la frontière du réseau permet de contrôler le flux du trafic entre le réseau local et Internet[40, 44]. En fonction des caractéristiques du trafic réseau, afin d'inclure les services demandés, les adresses de source et de destination et les utilisateurs individuels, un pare-feu décidera s'il convient ou non d'autoriser le trafic à transiter par le réseau. Les pare-feu peuvent également être utilisés sur des systèmes hôtes individuels[25].

Cependant, il y a des lacunes reconnues avec l'utilisation de pare-feu pour protéger un réseau.
Les défauts associés à un pare-feu sont les suivants :

1. Le pare-feu ne peut pas protéger contre les attaques qui le contournent.

2. Le pare-feu de l'interface réseau ne protège pas contre les menaces internes.

3. Le pare-feu ne peut pas protéger contre le transfert de virus déposé dans les fichiers et les programmes.

4. Dans certains cas, des volumes élevés de trafic réseau risquent de surcharger la capacité de surveillance réseau du pare-feu, ce qui pourrait entraîner le passage d'un trafic malveillant entre réseaux.

1.3.2 Système de détection d'intrusion (IDS) Un système de détection d'intrusion (IDS) détecte et alerte sur d'éventuels événements malveillants sur un réseau. Les capteurs IDS peuvent être placés à différents endroits du réseau. Un IDS est normalement basé sur une signature, c'est-à-dire qu'il recherchera les signatures prédéfinies des mauvais événements. Ces signatures ou règles résident normalement dans une base de données associée à l'IDS. Dès que l'ID détecte les signatures d'attaque, il met à jour les règles de filtrage du pare-feu. L'utilisation d'IDS et de pare-feu offre un niveau de sécurité élevé à l'administrateur système.

L'utilisation d'IDS en tant que dispositif de sécurité réseau présente également plusieurs inconvénients :

1. Un IDS basé sur un réseau doit pouvoir voir tout le trafic réseau du réseau qu'il protège. Si un réseau

utilise un commutateur (la plupart le font de nos jours), un sniffer ne pourra pas voir tout le trafic réseau. Cela signifie généralement que l'on déploierait un IDS réseau à la passerelle uniquement, c'est-à-dire sur sa connexion Internet. Cependant, cela ne le protège pas des attaques internes.

2. Les réseaux modernes sont si rapides qu'un système de détection d'intrusion a du mal à suivre.

3. Les IDS souffrent d'une surcharge de données. Ils ont tendance à générer un volume extrêmement important d'alertes. Ce volume rend l'analyse et la révision de toutes les alertes générées par le NIDS longues, coûteuses en ressources et en ressources. Par exemple, IDS déployé dans une organisation peut générer plus de 100 000 alertes par jour. Comme beaucoup de ces alertes sont de fausses alertes ou de faux positifs, les administrateurs commencent à ignorer la technologie et les alertes.

4. Il peut être difficile pour certaines technologies NIDS de détecter ou d'identifier des attaques ou des comportements inconnus. Cela laisse l'organisation vulnérable à de nouvelles attaques.

1.3.3 Système de prévention d'intrusion (IPS)

Les systèmes de détection d'intrusion (IDS), qui sont des mécanismes matériels et/ou logiciels, détectent et enregistrent des activités inappropriées, incorrectes ou anormales et les signalent aux fins d'enquêtes [5].De plus, le système de prévention d'intrusion (IPS) contient des fonctionnalités IDS mais des systèmes plus sophistiqués capables de prendre les mesures nécessaires pour prévenir ou réduire les activités malveillantes. [3].Lorsque les deux systèmes sont utilisés (IDS) et (IPS), on parle de système de détection et de prévention des intrusions (IDPS). Les systèmes de détection et de prévention des intrusions (IDPS) se concentrent sur l'identification des incidents éventuels, la consignation des informations les concernant, la tentative de les arrêter et leur communication aux administrateurs de la sécurité. En outre, les entreprises utilisent les IDPS à d'autres fins, telles que l'identification des problèmes liés aux menaces existantes et dissuadent les individus de violer les politiques de sécurité.

1.3.4 Pots de miel

Les pots de miel s'occupent de ces compromis et fournissent de petits ensembles de données.

1. Ils ne collectent des données que lorsque quelqu'un ou quelque chose interagit avec eux. Les organisations qui peuvent enregistrer des milliers d'alertes par jour ne peuvent enregistrer qu'une centaine d'alertes avec des pots de miel. Cela rend les pots de miel de données collectées beaucoup plus faciles à gérer et à analyser.

2. Les pots de miel réduisent considérablement les faux positifs. Toute activité impliquant des pots de miel est par définition non autorisée, ce qui la rend extrêmement efficace pour détecter les attaques.

3. Ils peuvent facilement identifier et capturer de nouvelles attaques contre eux. Toute activité avec le pot de miel est une anomalie, faisant ressortir facilement des attaques nouvelles ou invisibles.

4. Les pots de miel nécessitent un minimum de ressources, même sur le plus grand des réseaux. Un simple ordinateur Pentium peut surveiller littéralement des millions d'adresses IP.

5. Peu importe qu'une attaque soit cryptée, le pot de miel va capturer l'activité.

La stratégie derrière les pots de miel est de fermer les intrus des systèmes de production en toute sécurité et d'obtenir des informations sur les intrus en enregistrant leurs actions[62]. La figure 1.3 montre un exemple

de la manière dont des pots de miel peuvent être déployés dans un réseau à des fins de protection. Dans cet exemple, Honeypot-A simule un système sans pare-feu ni système de détection d'intrusion (IDS) ; Honeypot-B simule un service vulnérable tel que le service FTP ou Telnet pour attirer l'attention de Blackhat ; Honeypot-C et Honeypot-D simulent d'autres systèmes d'un réseau d'une organisation afin d'éloigner Blackhat des systèmes réels.

La plupart des pots de miel sont installés avec des pare-feu. La différence de le pare-feu et un pot de miel est que le pot de miel fonctionne dans le sens inverse. Il permet à tout le trafic d'entrer mais bloque tout le trafic sortant. La plupart des pots de miel sont installés à l'intérieur de pare-feu de réseau et permettent de surveiller et de suivre les pirates.

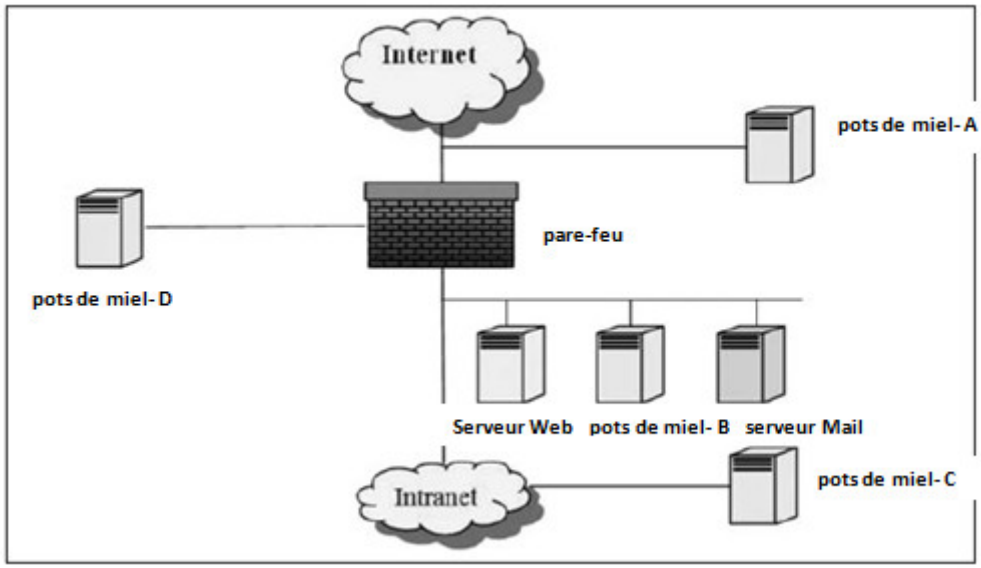

FIGURE 1.3 – Exemple de déploiement de pot de miel en réseau

[25]

1.4 Classification des pots de miel

Il existe plusieurs types de pots de miel, qui peuvent être regroupés en quatre grandes catégories[8].

1.4.1 Basé sur l'utilisation

1.4.1.1 Pots de miel de production

Un pots de miel de production est utilisé pour sécuriser un réseau opérationnel. Il déroute les attaques orientées vers les différents services de production du système, en les attirant vers lui, ce qui permet de réduire le risque, en renforçant la sécurité qui est assurée par les autres mécanismes de sécurité comme les firewalls, les IDS (Systèmes de Détections d'Intrusions), etc. Comme il peut aussi détecter des attaques grâce à ses fichiers d'audit, qui peuvent être aussi utilisés pour corriger les vulnérabilités.

Son rôle dans la protection du système

Un pots de miel de production joue un rôle important dans une ou plusieurs composantes de la sécurité du système de production telles que :

— **La prévention :** Laisser le hacker jouer sur le pots de miel au lieu de jouer sur les systèmes de production.

— **La détection :** Toute connexion établie avec un pots de miel de production est considérée comme tentative d'intrusion au système, il élimine ainsi toutes les fausses alertes (positives et négatives).

— **Le recouvrement :** Le rôle des pots de miel de production dans le recouvrement se traduit par les deux points suivants :

* Ils permettent une continuité des services après une attaque produite en leur sein, en les mettant simplement hors service.

* L'information enregistrée par les pots de miel de production sera d'un apport considérable pour le recouvrement du système.

1.4.1.2 Pots de miel de recherche

Le souci de ce type de pots de miel n'est pas de sécuriser un système particulier, mais c'est de s'introduire dans un environnement de recherche pour comprendre et étudier comment la communauté BlackHat évolue, quelles sont les techniques que cette communauté utilise et qui appartient à cette communauté. Les pots de miel de recherche sont plus complets que les pots de miel de production. C'est en général le système en entier qui peut être attaqué (et non pas seulement un seul service), ce qui en fait des systèmes sensibles dans leur gestion et complexes pour l'analyse de leurs résultats.

Son rôle dans la protection du système

Les pots de miel de recherche ne servent pas la sécurité des systèmes (prévention, détection et recouvrement) d'une manière directe, mais ils offrent des renseignements précieux sur les attaquants et leur comportement. Ces informations permettent une meilleure connaissance de la communauté des hackers, ce qui aide les professionnels de la sécurité informatique dans l'amélioration de méthodes et mécanismes de protection.

1.4.2 Basé sur le niveau d'interaction

L'implémentation d'un pots de miel est reposée principalement sur le niveau d'interaction1 "level of involvment" du pots de miel utilisé. Ainsi, nous pouvons distinguer trois classes différentes de pots de miel [spitzner2003honeypotstracking, spitzner2003honeypotsDefinitions] : les pots de miel à faible interaction, les pots de miel à moyenne interaction et les pots de miel à haute interaction.

1.4.2.1 Pots de miel à faible interaction

Un pots de miel à faible interaction est un pots de miel virtuel fournit comme le montre son nom une interaction limitée (faible) avec le pirate, il est tout simplement un programme qui simule les services d'un système réel[47] par la mise en place par exemple des sockets d'écoute sur chaque port d'un service, ces sockets ne font que logger les différents paquets qu'elle reçoit, comme le montre la figure 1.4.

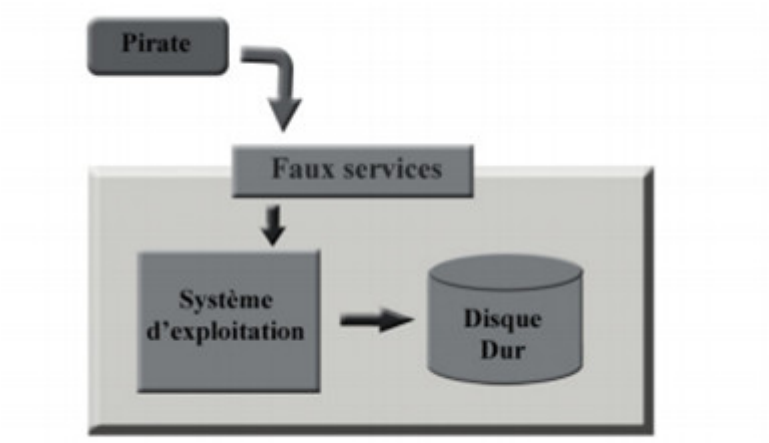

FIGURE 1.4 – Pots de miel à faible interaction

[8]

Les pots de miel à faible interaction les plus connus sont : Honeyd, Specter.

1.4.2.2 Pots de miel à moyenne interaction

Un pots de miel à moyenne interaction est un pots de miel semi-virtuel qui assure une simulation améliorée des services d'un système par rapport à la simulation fournie par les pots de miel à faible interaction, en lui ajoutant la possibilité de renvoi des réponses aux attaquants, ces réponses sont généralement fausses de façon à leur donner des pistes ou à les dérouter sans forcément les intriguer, comme le montre la figure 1.5. En plus des services simulés, il offre aussi quelques services réels, mais sans donner la possibilité au pirate de prendre un contrôle total du système.

Les pots de miels les plus connus de ce type sont : homemade honeypots, Deception Toolkit.

1.4.2.3 Pots de miel à haute interaction

Contrairement aux pots de miel à faible et à moyenne interaction, un pots de miel à haute interaction ne se base pas sur l'émulation d'un service, mais plutôt sur un vrai système d'exploitation, ce qui offre une grande interactivité avec l'attaquant, puisque il s'agit bien des systèmes réels avec des failles de sécurité qu'il peut exploiter, comme le montre la figure 1.6. Ce type de pots de miel est orienté plus à la recherche dont on souhaite qu'un pirate pénètre un système pour l'observer.

Les pots de miel à haute interaction les plus connus sont :Honeynet CDROM ROO, ManTrap

1.4.3 Basé sur l'architecture

L'architecture d'un pots de miel est définie par la nature du système qui l'héberge, qui peut être réel ou virtuel[28] :

1.4.3.1 Architecture réelle

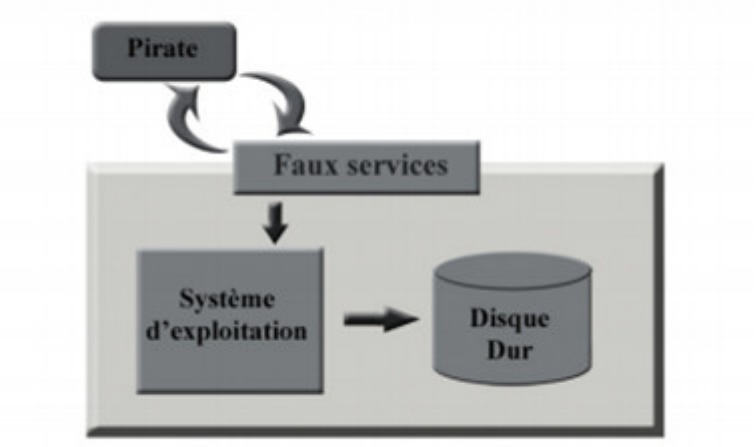

FIGURE 1.5 – pots de miel à moyenne interaction

[8]

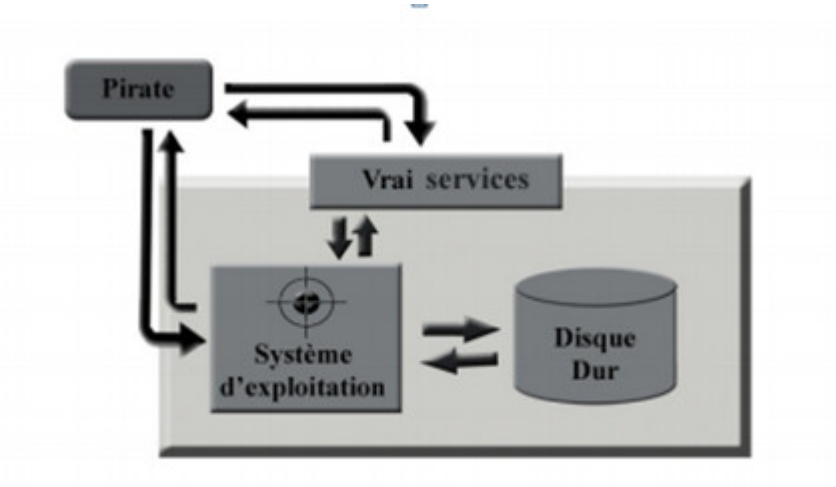

FIGURE 1.6 – pots de miel à haute interaction

[8]

Dans cette architecture, chaque pots de miel est installé sur sa propre machine physique, c'est-à-dire chaque pots de miel est représenté par un système réel.
Les avantages et les inconvénients de cette architecture sont :

— **Avantages :**

* Administration simplifiée.

— **Inconvénients :**

* Si plusieurs pots de miel alors plusieurs machines physiques.

* Le monitoring système sans se faire repérer par le pirate est compliqué.

* Réinstallation fréquente du système pour chaque pots de miel.

1.4.3.2 Architecture virtuelle

Dans cette architecture, le pots de miel est installé sur une machine virtuelle. La création des machines virtuelles est assurée par des outils de virtualisation de systèmes comme : VMWare[60] sous Linux et Windows, UML (User-Mode-Linux) [59] sous Linux, et Jail[26] sous Unix BSD. Ces outils peuvent émuler un ou plusieurs systèmes2 sur une seule machine, donc il est possible d'installer plusieurs pots de miel virtuels sur une seule machine. De plus, VMWare peut émuler plusieurs systèmes de natures différentes (windows, linux , ...etc), on peut alors proposer plusieurs pots de miel de plusieurs systèmes d'exploitation virtuels sur la même machine physique.
Les avantages et les inconvénients de cette architecture sont résumés dans :

— **Avantages :**

* Sécurité de la machine virtuelle.

* Economie de machines physiques.

* Possibilité de monitoring en temps réel des disques virtuels.

* Facilité de réinstallation par sauvegarde des disques virtuels.

* Le système hébergeant les systèmes virtuels est rendu invisible pour le pirate.

— **Inconvénients :**

* Charge importante du système hébergeant les machines virtuelles.

* Le choix du système virtuel est restreint à ceux qui sont compatibles.

1.4.4 Basé sur le rôle du pot de miel

— **Côté serveur :**

Les pots de miel conventionnels sont passifs de par leur conception et n'entraînent aucun trafic sauf s'ils sont compromis. Les pots de miel côté serveur sont utiles pour détecter de nouveaux exploits, collecter des logiciels malveillants et enrichir les recherches sur l'analyse des menaces. Exemples : pots de miel à faible interaction, honeyd, etc.

— **Côté client :**

Ce sont des pots de miel actifs pour les attaques côté client. Les attaques côté client représentent des attaques qui ciblent des applications clientes vulnérables, telles que des serveurs Web, lorsque le client interagit avec des serveurs malveillants. Le but de ces pots de miel est de rechercher et de détecter ces serveurs malveillants.
Strider Honey Monkey est un exemple de pot de miel côté client.

Conclusion

Comme nous avons expliqué dans ce chapitre, l'utilisation des pots de miel seul ne pas suffisent, elle ne remplace pas l'IDS mais peut être considérée comme un complémentaire et le point commun entre eux est de

capturer les attaquants et de collecter des informations précises.

Chapitre 2

État de l’art, étude comparative et travaux connexes.

Introduction

En général, le terme «pot de miel» est généralement utilisé pour désigner «un récipient (ou un pot) de miel». Mais dans le cas de la sécurité informatique, ce terme est utilisé pour représenter un concept de sécurité informatique basé uniquement sur la tromperie. Honeypot est une ressource pour piéger les outils et activités de l'attaquant. Lance Spitzner, fondateur de l'organisation The Honeynet Project, définit un honeypots comme suit : «Honeypot est une ressource de sécurité dont la valeur réside dans le fait d'être sondé, attaqué ou compromis». Cette définition indique la nature du pot de miel. Cela signifie que si personne n'attaque le pot de miel, ce n'est rien. Mais honeypot est un outil de sécurité précieux s'il est attaqué par l'attaquant. D'autres outils de sécurité, tels que le pare-feu et l'IDS, sont totalement passifs, leur tâche étant de prévenir ou de détecter les attaques. Honeypot donne activement à l'attaquant un moyen d'obtenir des informations sur les nouvelles intrusions. Cette nature rend honeypot exceptionnel pour aider d'autres outils de sécurité. Le pot de miel diffère selon l'utilisation différente. Il peut s'agir d'une application émulée, d'un système d'exploitation entièrement fonctionnel avec une configuration par défaut ou d'un réseau réel comprenant différents systèmes d'exploitation et applications, voire un réseau émulé sur une seule machine.

2.1 Systèmes de pots de miel

six pots de miel sont abordés dans la section suivante.

* ManTrap
* Back officer friendly
* Specter
* Honeyd
* Honeynet
* HoneyPy

2.1.1 ManTrap

ManTrap[33] est un pot de miel commercial à forte interaction créé, mis à jour et vendu par Recourse Technologies.

ManTrap crée un environnement d'exploitation hautement contrôlé avec lequel un attaquant peut interagir. Il crée un système d'exploitation entièrement fonctionnel contenant des cages virtuelles plutôt qu'un système d'exploitation limité. Les cages sont des environnements contrôlés logiquement dans lesquels l'attaquant est incapable de quitter et d'attaquer le système hôte. Cependant, au lieu de créer une cage vide et de la remplir avec certaines fonctionnalités, ManTrap crée des cages qui sont des copies miroir du système d'exploitation maître. Chaque cage est un système d'exploitation entièrement fonctionnel doté des mêmes fonctionnalités qu'une installation de production.

Cette approche crée une solution très puissante et flexible. Chaque cage est son propre monde virtuel avec peu de limitations. Un administrateur peut personnaliser chaque cage comme il le ferait avec un système séparé physiquement. Il peut créer des utilisateurs, installer des applications, exécuter des processus et même compiler ses propres fichiers binaires. Lorsqu'un intrus attaque et parvient à accéder à une cage, l'attaquant

donne l'impression que la cage est un système physique vraiment séparé. Il ne sait pas qu'il se trouve dans un environnement en cage où chaque action et frappe sont enregistrées.

2.1.2 BackOfficer Friendly (BOF)

BackOfficer Friendly[38], ou BOF comme on l'appelle communément, est une solution simple et gratuite pour pot de miel développée par Marcus Ranum. Il est extrêmement simple à installer, facile à configurer et nécessite peu de maintenance. Cependant, cette simplicité a un coût. Ses capacités sont sévèrement limitées. Il dispose d'un petit ensemble de services qui écoutent simplement sur les ports, avec des capacités d'émulation notamment limitées.

Cela fonctionne en créant des écouteurs de port, ou des sockets ouverts, qui se lient à un port et détectent toutes les connexions établies à ces ports. Lorsqu'une connexion est établie avec le port, les écouteurs de port établissent une connexion TCP complète (si le service est TCP), consignent la tentative, génèrent une alerte, puis ferment la connexion, en fonction de la configuration du service. Tout ce qui est BOF se passe dans l'espace utilisateur. Il ne crée ni ne personnalise aucun paquet lorsqu'il répond à des connexions. Grâce à ce modèle simple, BOF peut s'exécuter sur n'importe quelle plate-forme Windows, y compris Windows 95 et Windows 98.

2.1.3 Specter

Spectre[58] est un pot de miel supporté commercialement développé et vendu par les gens de NetSec. Comme BOF, Spectre est un pot de miel à faible interaction.

Cependant, Specter dispose de fonctionnalités et de capacités bien supérieures à celles de BOF. Spectre peut non seulement émuler plus de services, mais également émuler différents systèmes d'exploitation et vulnérabilités. Il dispose également de nombreuses fonctionnalités d'alerte et de journalisation. Étant donné que Specter n'émule que des services avec une interaction limitée, il est facile à déployer, facile à gérer et à faible risque. Cependant, comparé aux pots de miel à interaction moyenne et forte, il est limité dans la quantité d'informations qu'il peut collecter. Spectre est avant tout un pot de miel de production. Spectre partage les mêmes limites que BOF. En particulier, il ne peut ni écouter ni surveiller un port appartenant déjà à une autre application. Si un service écoute sur le port FTP (port 21), Spectre ne peut pas contrôler sur ce port. Spectre ne peut surveiller que les ports n'appartenant à aucune autre application. Il a également la capacité d'émuler différents systèmes d'exploitation. Ceci est fait en changeant le comportement des services pour imiter le système d'exploitation sélectionné.

2.1.4 Honeyd

Honeyd[42] est développé et mis à jour par Niels Provos de l'Université du Michigan et a été publié pour la première fois en avril 2002. Il est conçu comme une solution à faible interaction ; il n'existe aucun système d'exploitation destiné à un attaquant, mais uniquement des services émulés. Honeyd est conçu principalement comme un pot de miel de production, utilisé pour détecter des attaques ou des activités non autorisées.

Honeyd fonctionne sur le principe que lorsqu'il reçoit une sonde ou une connexion pour un système inexistant, il suppose que la tentative de connexion est hostile, très probablement une sonde, une analyse ou une attaque. Lorsque Honeyd reçoit un tel trafic, il assume l'adresse IP de la destination prévue (en le faisant victime). Il démarre ensuite un service émulé pour le port que la connexion tente. Une fois que le service émulé est démarré, il interagit avec l'attaquant et capture toute son activité. Lorsque l'attaquant a terminé, le service émulé

se ferme et n'est plus en cours d'exécution. Honeyd continue ensuite d'attendre le trafic ou les tentatives de connexion à des systèmes inexistants. Honeyd assume une adresse IP et exécute un service émulé uniquement lorsqu'il reçoit une tentative de connexion pour un système inexistant, méthode extrêmement efficace. Au fur et à mesure que Honeyd reçoit plus d'attaques, il répète le processus d'assimilation de l'adresse IP de la victime visée, en démarrant le service émulé respectif attaqué, en interagissant avec l'attaquant, en capturant l'attaque et en sortant. Il peut émuler plusieurs adresses IP et interagir avec différents attaquants en même temps.

2.1.5 Honeynets

Les honeynets[54] représentent l'extrême des pots de miel à haute interaction. Non seulement il offre à l'attaquant un système d'exploitation complet pour attaquer et interagir, mais il peut également fournir plusieurs pots de miel. Les Honeynets ne sont rien de plus qu'une variété de systèmes standard déployés dans un réseau hautement contrôlé. De par leur nature, ces systèmes deviennent des pots de miel, car leur intérêt est d'être sondés, attaqués ou compromis. Le réseau contrôlé capture toute l'activité qui se produit dans Honeynet et réduit le risque en limitant l'activité de l'attaquant

Les honeynets sont un mécanisme simple qui fonctionne sur le même principe qu'un pot de miel. Vous créez une ressource dont le trafic de production est faible ou nul. Tout ce qui est envoyé au Honeynet est suspect, potentiellement une sonde, un scan ou même une attaque. Tout ce qui est envoyé depuis un Honeynet implique qu'il a été compromis - un attaquant ou un outil est en train de lancer de l'activité. Cependant, Honeynets va encore plus loin dans le concept de pots de miel : au lieu d'un système unique, un Honeynet est un réseau physique de plusieurs systèmes.

Les Honeynets ne sont pas un produit que vous installez ou une appliance que vous déposez sur votre réseau. Honeynets est plutôt une architecture qui construit un réseau hautement contrôlé, dans lequel vous pouvez placer n'importe quel système ou application de votre choix.

2.1.6 HoneyPy

HoneyPy[32] ont été programmes avec de nombreux plugins. Le niveau d'interaction est déterminé par la fonctionnalité du plugin utilisé. Des plugins peuvent être créés pour émuler des services UDP ou TCP afin de fournir plus d'interaction. Toutes les activités sont consignées dans un fichier par défaut, mais vous pouvez également configurer l'activité de pot de miel sur Twitter ou sur un point de terminaison de service Web.

2.1.7 Les avantages et les inconvénients

Nom du pot de miel	Avantages	inconvénients
ManTrap	Fournit un mécanisme de réponse basé sur l'analyse de fréquence et arrête les machines en surveillant l'activité accrue des pirates informatiques. Fournit une surveillance furtive et donc une analyse d'attaque en direct. Détecte les intrusions sur l'hôte et sur le réseau	Besoin d'une expertise hautement qualifiée pour entretenir et déployer ces types de pots de miel. Même avec cela, le risque de compromission demeure et si ceux-ci sont connectés aux serveurs de production, une analyse de risque approfondie doit être effectuée.
BOF	Facile à installer, configurer et entretenir. Fonctionne sur toutes les plateformes Windows ou Unix. Peu de risque en raison de la simplicité.	Limité à sept ports sur lesquels il peut détecter des attaques. Les ports ne peuvent pas être personnalisés, ce qui augmente les possibilités d'empreintes digitales. Pas de journalisation à distance, d'alerte ou de configuration de la personnalité.
Specter	Facile à installer, configurer et déployer. Émulation de service étendue. Surveille deux fois plus de ports que BOF. Capacités de notification en suspens.	Surveille seulement 14 ports. Les services émulés préprogrammés sont limités à une interaction avec un comportement connu. Limitations concernant les informations collectées, principalement les informations transactionnelles et l'interaction de l'attaquant avec les sept services émulés.

Honeyd	Peut surveiller n'importe quel port TCP ou UDP et des réseaux entiers. En tant que solution open source, il est gratuit et se développera rapidement avec le contribution et développement des autres dans la communauté de sécurité. Résistez aux efforts d'empreinte digitale en émulant le système d'exploitation sur une pile IP niveau.	En tant que solution à faible interaction, elle ne peut pas fournir un véritable système d'exploitation avec lequel les attaquants peuvent interagir. En tant que solution open source, il ne fournit aucune assistance formelle pour la maintenance et le dépannage. Pas de mécanisme d'alerte intégré
Honeynets	Flexibilité, tout système peut être placé dans l'environnement Honeynet. Capacités de capture de données étendues pour les outils et les tactiques connus et inconnus. Adaptable à de nombreuses organisations et environnements.	Complexité du déploiement et des ressources nécessaires à la maintenance. La fonctionnalité d'interaction élevée introduit le risque que des attaquants utilisent les systèmes pour attaquer ou nuire à un autre système. Les technologies nouvelles et immatures ont un risque plus grand de se casser et d'introduire des erreurs.
HoneyPy	HoneyPy est Facile à : installer et déployer étendre avec des plugins et des enregistreurs exécuter avec des configurations personnalisées Solution open source, et gratuit	

TABLE 2.1 – Les avantages et les inconvénients des pots de miel.

2.1.8 Comparaison entre les différents pots de miel

Dans cette section, les cinq pots de miel sont comparés sous forme de tableau.

* Le niveau d'interaction entre l'utilisateur et le pot de miel est élevé dans le cas de Mantrap, spectre et Honeynet et ce niveau est faible dans le cas de la fonction BF et de la propriété.

* Honeyd et Honeynet sont disponibles gratuitement, alors que Mantrap, spectre et BOF ne le sont pas.

* Honeyd et Honeynet sont des logiciels open source, alors que Mantrap, Spectre et BOF ne le sont pas.

* BOF ne supporte pas le fichier journal alors que le reste des pots de miel supporte le fichier journal.

* BOF n'émule pas le système d'exploitation alors que le reste des quatre pots de miel peut émuler le système d'exploitation.

* Les services illimités sont pris en charge par ManTrap, Honeyd et Honeynet, tandis que les services limités sont pris en charge par le BOF et le spectre.

-	**ManTrap**	**BOF**	**Specter**	**Honeyd**	**Honeynet**	**HoneyPy**
Niveau d'interaction	Haut	Faible	Haut	Faible	Haut	faible à moyen
Disponible gratuitement	Non	Non	Non	Oui	Oui	Oui
Open source	Non	Non	Non	Oui	Oui	Oui
Support de fichier journal	Oui	Non	Oui	Oui	Oui	Oui
Emulation OS	Oui	Non	Oui	Oui	Oui	Oui
Services pris en charge	Libre	7	13	Libre	Libre	Libre

TABLE 2.2 – Comparaison entre les différents pots de miel :.

2.2 Traveaux connexes

Les systèmes Honeypot ne sont utilisés ni pour détecter le système de détection d'intrusion, ni le pare-feu pour un problème spécifique direct. Les pots de miel font partie des systèmes de sécurité et le type de problème qu'ils vont résoudre dépend de la conception et de l'utilisation. Par conséquent, au contraire d'autres équipements de sécurité de l'information, il n'est pas possible de mentionner un pot de miel capable de donner une réponse générale à chaque solution de problème [20, 19]. Dans la littérature technique, il existe diverses applications de sécurité telles que la détection et la prévention des intrusions (IDPS) qui sont utilisées collectivement.

Riboldi et al. ont mis au point un système honeypot à faible interaction pour surveiller les activités illégales sur les systèmes VOIP dans leur étude. Pendant 92 jours sur le système dont les performances ont été surveillées, 3502 événements liés au protocole SIP ont été rassemblés. Ils ont interprété leur système comme étant disponible, comme un environnement VOIP de pare-feu et de système de détection d'intrusion[45]. Shukla et al. ont mis en place un système cher pour détecter les URL Web malveillantes dans leurs études. Le système développé par le langage Python est servi côté client. À l'aide d'un robot d'exploration côté client, les

adresses URL sont rassemblées et les sites Web sont consultés par la suite. Si ces URL sont malveillantes ou contiennent une vulnérabilité du système de détection d'intrusion basé sur la signature, un déclencheur est activé. Ainsi, les adresses URL malveillantes sont enregistrées dans la liste noire et la sécurité est disponible[51].

Koniaris et al. ont utilisé des systèmes honeypot pour l'analyse et la visualisation d'activités et de connexions malveillantes. Dans leur application exécutée, ils ont mis en place deux autres pots de miel de recherche. Le premier d'entre eux, généralement doté d'une option d'auto-propagation, a été conçu pour rassembler des logiciels malveillants et le second, pour regrouper des activités malveillantes en tant que système de piège[27]. Song Li et al. ont délibérément mis en place un système de détection d'intrusion basé sur un creuset à interactions mixtes. Ils expliquent le but du système qu'ils ont mis au point pour stabiliser le réseau et améliorer la sécurité. En raison de l'amélioration de la sécurité du réseau, ils ont augmenté la capacité de piège des systèmes à pots de miel et ont mené diverses recherches[30]. Chawda et al. ont proposé un système distribué de pots de miel pour rechercher de nouvelles vulnérabilités. Dans leur système exécuté pour être exposés à une vulnérabilité supplémentaire en tant que filtre de contenu frontal, ils ont utilisé des systèmes de pots de miel à faible interaction[10].

Xiangfeng Suo et al. ont délibéré sur la manière de mettre en pratique les technologies du pot de miel dans les systèmes de détection d'intrusion. Dans le cadre de travaux de recherche, ils ont soumis une proposition visant à mettre en pratique les systèmes à pots de miel afin de résoudre les problèmes liés au système de détection d'intrusion[56]. Paul et al. ont effectué un générateur de signature basé sur un pot de miel pour la sécurité du réseau informatique. Le système développé a été spécialement utilisé dans le but de se protéger contre les attaques de vers polymorphes. Le système développé est également capable d'isoler le trafic suspect et de rassembler de nombreuses données utiles sur le trafic malveillant et les attaques de vers. Lorsque les systèmes basés sur la signature ne fonctionnent pas pour détecter de nouvelles attaques pour les attaques par vers inconnues, il est habilité à générer une signature[39].

Beham et al. ont bénéficié des avantages des technologies de virtualisation. Dans leur étude, ils ont étudié la détection d'intrusion et l'environnement de virtualisation imbriqué de systèmes honeypot. Dans l'étude, les technologies de virtualisation imbriquées actuelles, la détection d'intrusion et les systèmes de pots de miel basés sur VMI ont fait l'objet d'une recherche comparative[6]. Liu et al. ont mis en place un système de détection d'intrusion, basé sur le principe du pot de miel et utilisant la technique de traçage IP. Pour introduire les limites des systèmes de détection d'intrusion classiques sur les systèmes à pots de miel, une conception de détection d'intrusion a été proposée[12]. Dans son étude, Auttopan Pomsathit a traité de l'utilisation des systèmes de pots de miel et des systèmes de détection d'intrusion sur des réseaux distribués. Il a expliqué que son objectif principal était de mesurer l'efficacité des systèmes de détection d'intrusion en combinant des systèmes de détection d'intrusion et des pots de miel[4].

Jiang et al. ont géré l'application système du pot de miel pour les réseaux d'entreprise. Ils ont combiné

les méthodes utilisées dans les systèmes de détection d'intrusion avec un nouveau système honeypot afin de visualiser les systèmes honeypot actuels[17]. Mitsuaki et al. ont conçu un honeypot client évolutif à haute interaction et performances efficaces. De cette manière, une analyse en profondeur et une capacité de capture ont été ciblées[1].

P.Fanfara et al. se sont concentrés sur la technologie appelée honeypot et sur la question du processus de mise en œuvre de sa version autonome, qui est capable de créer des pots de miel virtuels et donc d'accroître rapidement le niveau de sécurité des systèmes informatiques hétérogènes distribués dans leur étude[14]. Markert J. et al. ont présenté une analyse efficace d'un pot de miel pour WSN et ont présenté des capacités de détection dans les catégories d'attaques connues et inconnues[34]. Musca C. et al. ont présenté des méthodes pour isoler le trafic malveillant en utilisant un système honeypot et en l'analysant afin de générer automatiquement des signatures d'attaque pour le système de détection / prévention d'intrusion SNORT dans leur étude[37]. Sadasivam G.K. et al. ont déployé plusieurs pots de miel dans un environnement virtualisé pour rassembler des traces d'activités malveillantes sur leur papier[48].

Conclusion

Les pots de miel sont les ressources de sécurité pouvant contribuer à la sécurité du réseau. Différents systèmes de pots de miel ont été discutés et également on été comparer les différents systèmes. Chaque pot de miel a ses avantages et ses inconvénients. Différents systèmes de pots de miel peuvent être déployés dans différentes conditions. Un administrateur peut choisir un pots de miel en fonction de ses besoins.

Chapitre 3

Contribution.

Introduction

Comme nous l'avons expliqué dans la section précédente, Les pots de miel et les SPI sont d'excellents moyens pour améliorer la sécurité interne d'un système d'information. Ils peuvent également constituer une menace pour les performances et la productivité des SI. Afin de compléter l'opération des pots de miel afin de les fabriquer et de réduire les risques sur le SI, nous allons présenter l'architecture globale de ce nouveau IPS nommé IPMS (Intrusion Prevention Management System) et examinerons les différentes façons dont il peut interagir avec des périphériques réseau pour sécuriser le IS.

3.1 Architecture distribuée pour la prévention d'intrusion basée sur honeyd

Dans le but de compléter le fonctionnement des pots de miel pour les faire évoluer vers un fonctionnement réactif, et d'en sortir avec un nouveau IPS distribué nommé IPMS (Intrusion Prevention Management System), sûr et avec un minimum d'impact et de risque sur le SI, nous allons présenter l'architecture globale pour ce nouveau IPS et nous étudierons les différentes manières avec lesquelles il peut interagir avec les équipements réseau pour sécuriser le SI. Les principaux objectifs étant de :

1. Palier aux faiblesses des IPS réseau standards, à savoir :
   a. corriger l'aspect « maillon faible ».
   b. être invisible dans le réseau du SI.
   c. réduire le taux de faux positifs qui nuisent au rendement et à l'efficacité de tout IPS.
2. Avoir un système réparti dans tous les segments d'un réseau de SI.
3. Ne plus avoir de goulet d'étranglement dans le réseau.
4. Avoir un IPS qui s'intègre dans un réseau avec zéro impact sur l'existant.
5. Avoir un IPS avec un système de décision adaptable à la nature du métier sans changer toutes les composantes d'un SI.

Dans ce qui suit, nous allons commencer par présenter l'architecture proposée de notre architcture proposé de l'IPHBS [1], ensuite nous présenterons le cycle d'amélioration que nous avons suivi dans le processus de mise en place, et enfin nous présenterons l'architecture finale avec les résultats obtenus.

3.1.1 Architecture fontionnelle de notre système

Nous abordons dans cette partie la solution de prévention d'intrusion proposée comme réponse aux différentes limitations des plateformes classiques décrites auparavant.

1. Intrusion Prevention Honeypots Based System

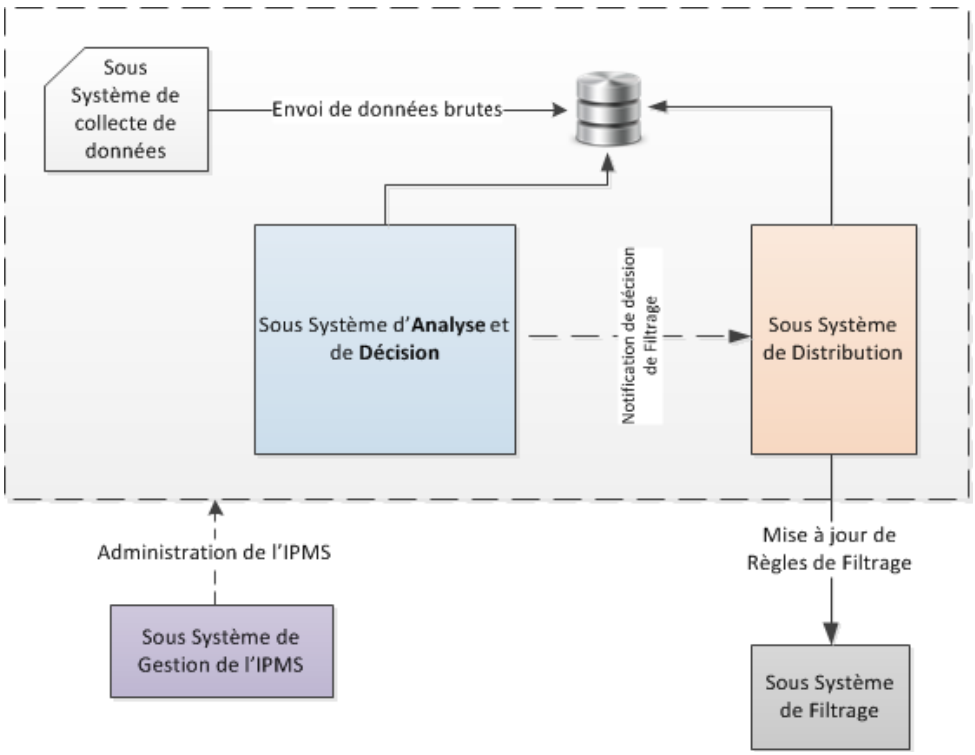

FIGURE 3.1 – Architecture fonctionnelle de l'IPMS

3.1.1.1 Description Générale

L'objectif principal de notre architecture est de pouvoir prévenir les intrusions d'une manière automatique en minimisant autant que possible les faux positifs. Cette architecture est composée globalement de cinq sous-systèmes :

1. un sous-système de collecte de données,
2. un sous-système d'analyse et de décision
3. un sous-système de distribution
4. un sous-système de filtrage
5. et un sous-système de Gestion de notre système lui-même.

La figure 3.1 présente une vue générale de notre systeme.

Toute la plateforme de notre système se base sur l'analyse des données collectées par le sous-système de collecte de données d'une manière centralisée au niveau du sous-système d'Analyse et de Décision pour générer des alertes en cas de détection d'une anomalie. Ces alertes sont codées en règles de sécurité par le sous-système de distribution selon un format adapté au sous-système de filtrage.

3.1.1.2 Contexte et hypothèses de fonctionnement

La plateforme en étude n'est pas en elle-même un outil de filtrage. Ceci implique que nous faisons l'hypothèse que le réseau à protéger est déjà muni d'outils de filtrage tels que routeurs supportant les fonctionnalités

de filtrage, des pare-feux matériels ou logiciels. Notre systeme vient compléter et aider un système de sécurité déjà établi pour lui apporter de l'autonomie de décision, de la prévoyance ainsi qu'un autre niveau de sécurité avec un minimum d'impact sur la configuration de l'existant.

Ainsi, Notre systeme ne peut pas, et n'est pas destiné à remplacer les mécanismes de filtrage classiques mais à les renforcer.

Notre systeme offre aussi, grâce aux fonctionnalités natives de Honeyd, la possibilité d'étendre cette architecture de sécurité à travers un WAN pour avoir une vision globale sur l'état de la sécurité de tout le réseau d'entreprise.

3.2 Data mining

Data mining signifie l'extraction de connaissance a travers l'analyse d'une grande quantité de données pour utiliser ces connaissances dans le processus de décision. On peut trouver les données stocker et organisé dans les datamarts et les entrepôts de données, ou dans autre sources non structurées. Le processus de data mining implique plusieurs étapes avant de trouver un modèle de décision qui peut être un ensemble de règles, des équation ou des fonction de transfert complexes. Dans notre travail nous avons deux étapes : la collecte de données et l'analyses et décision. Ce deux étapes sont répartis sur deux sous-systèmes. Selon leur objectif le data mining se compose en deux catégories supervisé et non supervisé. Nous travaillons sur les données d'une façon non supervisé, car elle est plus rapide et facile à mettre en place.

3.2.1 Sous-système de collecte de données La collecte des toutes les informations sur les connexions suspectes ainsi que leurs caractéristiques (adresse IP source, port source, adresse IP de destination, port de destination, contenu du paquet... etc.) se fait au niveau de ce sous système. Concrètement, pour notre architecture, il s'agit des pots de miel Honeyd transformés en « sondes ». Ces sondes sont partout dans le réseau là où l'administrateur de sécurité juge nécessaire d'en avoir. Concrètement, quelques emplacements réseau sont très communs d'usage, d'autres restent à l'appréciation de l'administrateur.
Les emplacements les plus utilisés sont :

* Nœuds d'accès à Internet : la sonde est connectée aux équipements de connexion à tout réseau externe en général et Internet en particulier.

* La DMZ : la zone démilitarisée est le segment réseau où sont connectés les serveurs hébergeant des services destinés à un accès public; par exemple : le serveur Web de l'entreprise, le serveur de messagerie... etc. Une sonde pourrait être placée parmi ces serveurs pour pourvoir observer tout comportement anormal.

* Sur le réseau des serveurs internes (serveurs de stockage, de base de données principalement) qui sont de cibles très convoitées dans le cas des attaques de vol d'information.

* Sur le réseau interne de l'entreprise; des études ont montré qu'un grand pourcentage des attaques proviennent de l'interne de l'entreprise, soit d'une façon préméditée dans le cas d'un employé mécontent ou agissant pour un concurrent, soit sans même que l'utilisateur soit conscient surtout dans le cas des utilisateurs

nomades avec des PC portatifs infectés par des chevaux de trois par exemple.

Ce sous système de collecte de données communique avec le sous-système d'analyse et de décision pour alimenter la base de données de Notre système.

3.2.1.1 Sous-système d'analyse et de décision

Quel que soit l'outil de protection d'un système, il est constitué de deux parties :

* une partie que nous pouvons qualifier de technique, c'est la partie qui va prendre l'information utile au système d'analyse pour prendre sa décision. Par exemple, pour un antivirus, ça pourrait être une signature d'un fichier, une chaine de caractère dans un fichier . . . etc. pour un système de détection d'intrusion de type hôte, ça pourrait être la détection d'une action, ou suite d'actions, sur le système pour un exécutable donné. . . etc.

* une partie analyse et décision : c'est cette partie qui fera toujours la différence entre un outil et un autre. La fiabilité du moteur d'analyse, sa rapidité, les algorithmes de décision. . . etc. feront qu'un outil est plus stable et fiable qu'un autre sur un système (ou famille de système) donné.

Dans le cas du sous-système d'analyse et de décision assure, comme son nom l'indique, son rôle est l'analyse des données collectées ainsi que la prise de décision sur la légitimité d'un trafic.

C'est le maillon intelligent de Notre systeme. Où nous avons implémenté méthode de décision par Partitionnement en utilisant l'algorithme de K-means sans que cela impacte Notre systeme en terme d'architecture, mais nécessairement en terme de degré d'exactitude et de diminution du taux de faux positifs.

Une fois que la décision est prise sur la nature d'un trafic, soit que c'est un trafic légitime et auquel cas rien n'est fait. Ceci est un cas rare puisqu'aucun trafic légitime ne devrait atteindre les sondes, ce qui veut dire que nécessairement et à 99,999% des cas c'est un trafic qui n'a pas lieu à exister. La seule exception à cette règle pourrait être un administrateur qui scanne tout le réseau pour faire un recensement de la plateforme des serveurs (voir quels ports sont en état d'écoute, quelles adresses IP sont utilisées . . . etc.) et auquel cas aucune action ne sera entreprise.

Donc si nous sommes dans le scénario d'un trafic malicieux, le sous-système de décision va générer une requête vers le sous-système de distribution avec les informations nécessaires pour qu'il génère les règles de filtrage pour tous les équipements réseau concernés.

3.2.1.2 Partitionnement

Construire une partition de la base de données D contenant n objets en un ensemble de k clusters.
Étant donné k, trouvé une partition en k clusters qui optimisent le critère de partitionnement.

Optimum global : traiter toutes les partitions exhaustivement
Heuristique : k-means ou k-médoïdes
k-means : chaque cluster est représenté par son centre
k-médoïdes ou PAM (partition around medoids) : chaque cluster est représenté par un des objets du cluster.

3.2.1.3 Sous-système de distribution

Une fois le sous-système de décision soumet une nouvelle demande de blocage de trafic, le sous-système de distribution adapte la demande selon la nature du sous-système de filtrage destinataire. Par exemple, une règle iptables sur un serveur linux ou une access-list pour un routeur ou firewall. Le sous-système de distribution tel qu'il a été pensé est adapté aux plateformes de la famille Unix ou équipements tolérant un accès distant avec le protocole Telnet ou SSH. Cette famille étant la plus répondue (linux, Unix, Microsoft Windows, routeurs et Firewalls des constructeurs majeurs) le plus grand rôle du sous-système de distribution reste alors l'adaptation de syntaxe selon le sous-système de filtrage ciblé.

3.2.1.4 Sous-système de filtrage

Un sous-système de filtrage peut être n'importe quel système réalisant les fonctionnalités de filtrage IP et supportant une configuration à distance en utilisant Telnet ou SSH ou tout autre mécanisme équivalent.

Les équipements tiers et les systèmes d'exploitation offrent tous un accès distant permettant ce genre de configuration. Par ailleurs, la question du choix du périphérique sur lequel une règle s'appliquera reste posée. Pour ceci il y a deux aspects importants à prendre en compte : est-ce que l'équipement de filtrage est frontalier ou dorsal, et si le réseau à protéger, ou plage d'adresse IP à filtrer, est dans la juridiction du périphérique. En règle générale, si la sonde qui a détecté l'anomalie est dans le réseau publique ou est dans la zone démilitarisée (DMZ) il est judicieux d'appliquer la règle générée sur tous les équipements de périmètre susceptibles de recevoir l'attaque. Si par contre nous estimons que la cible de l'attaque observée est derrière un parefeu dorsal, là il faut voir l'opportunité d'ajouter une règle à ce niveau aussi.

D'un autre côté, si la sonde détecte une attaque provenant du réseau interne, il sera nécessaire d'effectuer un blocage à tous les niveaux car en général un trafic « sortant » n'est jamais bloqué et là il sera en particulier nécessaire de le faire. Un administrateur qui observe une telle situation devrait faire une recherche approfondie sur l'hôte qui a généré l'attaque car il est susceptible de contaminer ses paires s'il est laissé sans correction du problème.

3.2.1.5 Sous-système de gestion de notre système

Ce sous-système n'a aucun impact sur le fonctionnement de Notre système mais il permet, d'une façon centralisée, d'administrer et gérer la plateforme, à savoir, la manipulation des sondes (ajout, modification et

désactivation) ainsi que des règles automatiques et ainsi de suite. Nous aborderons plus en détails ses fonctionnalités dans le chapitre qui suivra.

Conclusion

Dans le chapitre "Contribution" qui est l'avant dernière partie dans notre travail, nous avons présenté les fonctions principales de notre projet, autrement dit, les avantages offerts par le système et comment manipuler les différents composant

Chapitre 4

Étude expérimentale.

Introduction

Apres avoir décrit le role et fonctionnement de chaque sous-systeme du IPMS, dans ce chapitre nous allons présenter l'implémentation de notre systeme. Nous commençant par présenter la platforme utilisé après nous détaillons l'implémentation de chaque sous-système.

Platforme

Nous avons choisis de travailler sur Graphical Network Simulator-3 (GNS3) [53]. GNS3 est un émulateur de logiciel de réseau publié en 2008[22, 9, 11]. Il permet la combinaison de périphériques virtuels et réels, utilisés pour simuler des réseaux complexes. GNS3 est utilisé par de nombreuses grandes entreprises, notamment Exxon, Walmart, AT&T et NASA, et est également très apprécié pour la préparation d'examens de certification professionnelle en réseau. En 2015, le logiciel avait été téléchargé 11 millions de fois [15]. Il est sous licence GPL-3.0, et fonctionne sous tous les systèmes d'exploitation. GNS3 nous permet d'installer nos sondes (honeyd) sous linux Ubuntu server comme montré dans la figure 4.1, et notre serveur sous-systeme de gestion et serveur de récolte sur Linux ubuntu desktop.

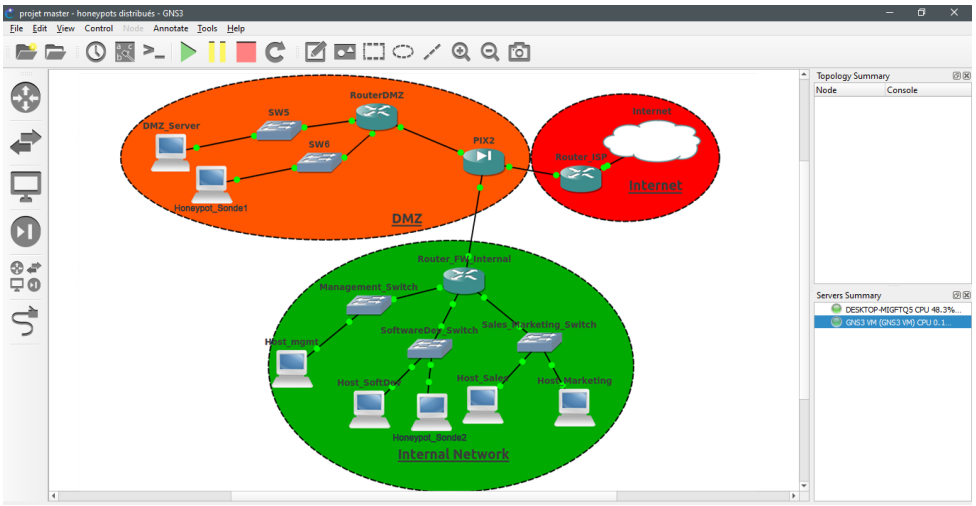

FIGURE 4.1 – Interface graphique du GNS3

4.1 Sous-système de collecte de données

Comme nous l'avons présenté, ce sous-système est le point d'entrée de toutes les données collectées. Il est dans notre cas représenté par un pot de miel basé sur Honeyd. Nous appellerons le serveur hébergeant le pot de miel sonde dans la suite de ce document. Chaque sonde peut simuler un ou plusieurs services et envoi les données qu'elle collecte vers le sous-système d'analyse et de décision. En pratique, les sources de Honeyd ont été modifiées pour ajouter une interface avec MySQL [1] et pouvoir ainsi corréler les données provenant de toutes les sondes du réseau.

1. Système de Gestion de Base de Données libre sous licence GPL, et a été racheté par Oracle.

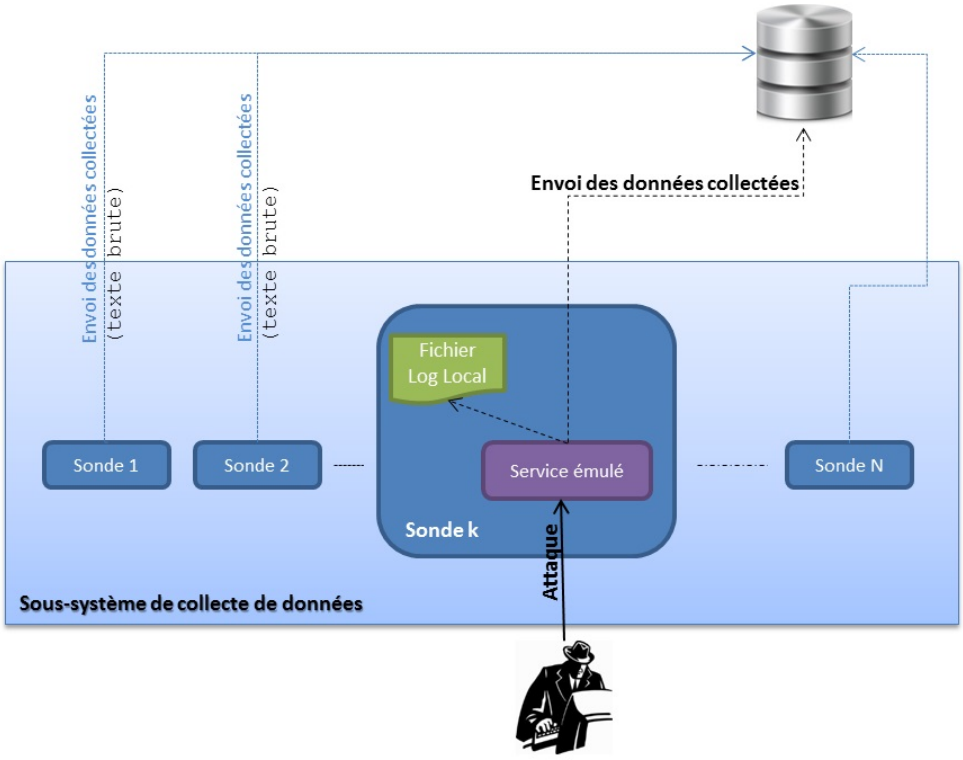

FIGURE 4.2 – Sous-système de collecte de données

Cet aspect distribué de la localisation des sondes, ainsi que leur multitude, permet d'avoir beaucoup d'informations sur ce qui se passe sur le réseau et les différentes menaces auxquelles il doit faire face. Ceci est d'autant plus important que les réseaux actuels sont de plus en plus compliqués en termes d'architecture (segmentation réseau, sites géographiquement distants...etc) et en terme de technologies utilisées. En effet, nous pouvons avoir plusieurs architectures réseaux d'entreprise et une multitude d'équipements réalisant les fonctionnalités de filtrage à différents niveaux :

— zone démilitarisée (DMZ) : contient les serveurs hébergeant des services destinés à être consultés de l'extérieur de l'entreprise (Internet). Ces serveurs ne vont, en général, jamais initier de connexion vers l'intérieur du réseau de l'entreprise.

— réseau frontal (FE) : dans une architecture basée sur des configurations de services à deux niveaux frontal et dorsal, ce réseau contient les serveurs accessibles de l'extérieur. Par exemple, on peut avoir des serveurs frontaux SMTP qui relaient lesemails vers des serveurs contenant les boites emails des utilisateurs dans le réseau dorsal après analyse antivirale.

— réseau dorsal (BE) : ce sont des serveurs qui ne sont pas accessibles de l'extérieur. Selon la politique de sécurité de l'entreprise, on peut aussi interdire l'accès même aux utilisateurs internes pour les obliger à passer par le réseau FE.

— réseau d'utilisateurs (userLan) : c'est le réseau qui contient des utilisateurs normaux qui exploitent les ressources de l'entreprise pour en sortir une valeur pour cette dernière. C'est aussi le réseau qui contiendra les utilisateurs plus ou moins familiarisés avec les systèmes informatiques, autrement dit, différents

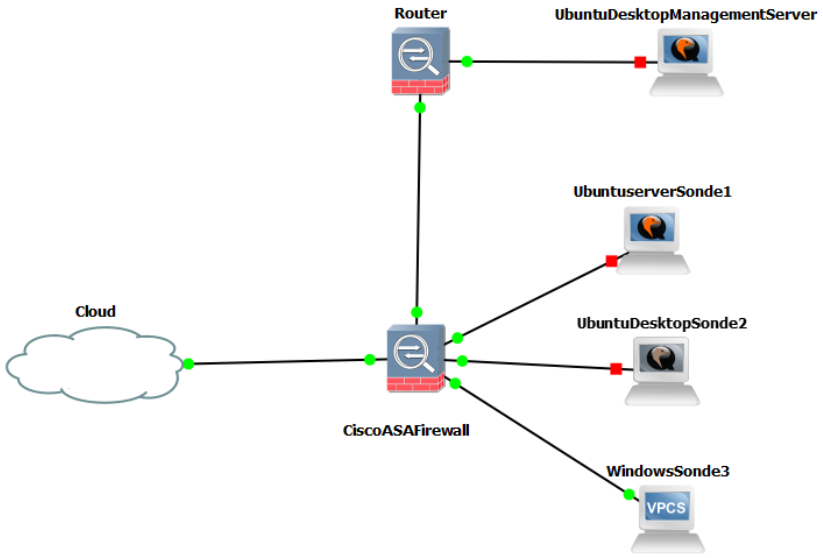

FIGURE 4.3 – Expérimentation : le réseau de machines virtuels construit sur GNS3

niveaux de risque de fraude.

— réseau de données (dataLan) : en général ce réseau n'est jamais accessibles directement aux utilisateurs mais seulement aux serveurs BE. Dans le passé ces réseaux étaient en général des réseaux à fibre optique dédiée au transport des données brutes (raw) entre serveurs et baies de stockage à l'aide du protocole SCSI [2] par exemple. Mais actuellement, on peut avoir de tels protocoles même sur IP (le iSCSI, le i pour IP), ce qui introduit un risque de sécurité de données.

Ces réseaux restent les plus communs et tout autre réseau pourrait être assimilé à l'un d'entre eux.//

Dans nos expérimentations nous avons mis en place des machines virtuels sondes dotés de système d'exploitation Linux Ubuntu server. Où nous avons installé et configuré Honeyd pour faire la capture. Nous avons aussi programmé un client avec Java qui communique avec le sous-systeme d'analyse et de décision qui est au Management server 4.3. Ce dernier envoie les fichiers log de capture générés par Honeyd par trois façons :

— envoie périodique : Chaque période de temps initialement égale à une heur, la sonde envoie le fichier de capture si il a changé. La sonde double le temps de capture avant envoie si il n'y a eu aucune capture. La période d'attente et l'option du temps adaptatifs peuvent être configuré par le sous-système de gestion.

— envoie par demande : dans ce cas la sonde est en mode standby jusqu'en que la sonde reçois une demande d'envoi par le sous-système de gestion.

— envoie par seuil : finalement dans ce cas la sonde compte le nombre d'entrés dans le fichier log généré par le honeypot, si le nombre atteint un seuil déterminé par le sous-système de gestion, la sonde envoie les captures au sous-système d'analyse et de décision.

2. SCSI : Small Computer System Interface. C'est un ensemble de standards pour lier physiquement et transférer les données entre serveurs et autre périphériques de stockage

4.2 Description du sous-système d'analyse et de décision

Comme nous l'avons expliqué, le sous-système d'analyse et de décision fait l'analyse des informations envoyées par les sondes pour décider si oui ou non il s'agit d'une attaque et éventuellement déclencher des actions en conséquence. C'est le sous-système le plus important dans la chaîne de notre système du moment où c'est à ce niveau que nous pouvons analyser et corréler les évènements remontés par les sondes avec les règles de gestion du système d'information ainsi que d'autres informations aidant à la sécurité du SI. La méthode d'analyse se base principalement sur les caractéristiques des connections et des types de données collectées pour évaluer un trafic donné. En effet, vu que le système n'est pas basé sur la détection d'intrusions cataloguées et identifiées par leurs signatures, le choix des caractéristiques du trafic aidant à la prise de décision impacte grandement la décision finale.

Pour améliorer le système d'analyse, le mécanisme de listes blanche et noir a été implémenté 4.4. L'idée est d'avoir une liste avec les adresses des sources de trafic qui ne doivent être bloquées car ceci se traduirait par un arrêt de service, cette liste c'est la liste Blanche. La liste noire représente quant à elle des adresses communément [3] reconnues comme source d'attaques. En effet, ceci permet de réduire le taux des faux positifs, d'avoir une liste des sources d'attaques connues mais aussi d'éviter tout arrêt de service si jamais le réseau subis un grand trafic lors de périodes de forte charge ou si jamais un attaquant arrive à usurper l'adresse IP d'un serveur de production. Dans [36] les auteurs s'intéressent à la construction dynamique de cette liste noire en utilisant des coefficients de confiance par adresse IP. Les auteurs proposent de faire une analyse des paquets provenant des adresses IP de confiance avant qu'ils n'atteignent même le nIPS. Si une source dans la liste noire usurpe une adresse IP de confiance, les auteurs affirment empiriquement que le taux de faux positifs et négatifs reste inchangé grâce à la méthode d'analyse statistique des paquets. Autrement dit, aucun impact direct de l'usurpation d'IP sur le fonctionnement global du nIPS.

Il est d'ailleurs possible d'améliorer tout le système si un mécanisme de signature est utilisé. Le greffon HoneyComb [52] permet de générer des signatures uniques basées sur l'analyse des données collectées par Honeyd. Dans [41] les auteurs ont établi une architecture de détection de prolifération de vers informatiques en générant automatiquement leurs signatures en utilisant HoneyComb. Cette plateforme est gérée d'une manière centralisée à travers un Centre de Contrôle. Le principe de liste blanche est aussi utilisé dans cette plateforme pour minimiser le taux de faux positifs comme déjà discuté dans le paragraphe précédent. Nous ne nous intéresserons pas à la génération des signatures dans ce travail vu qu'elle peut être ajoutée à n'importe quel moment au système. Dans notre système nous avons utilisé une méthode d'apprentissage non supervisé de classification pour catégoriser si le trafic doit être ajouté à la liste noire ou dans la liste blanche.

4.2.1 Description de la méthode de décision

Le choix de la méthode de décision a un grand impact sur les résultats qui seront obtenus par la suite. En effet, prendre une méthode non adaptée à la nature des données et de toute la plateforme pourrait générer des résultats inattendus et totalement à l'encontre de ce que nous voulons réaliser, ou tout au moins avec un grand degré d'inexactitude.

3. Ces listes sont en général construites par des éditeurs de solutions de sécurité ou bien par une communauté d'acteurs dans le domaine de sécurité

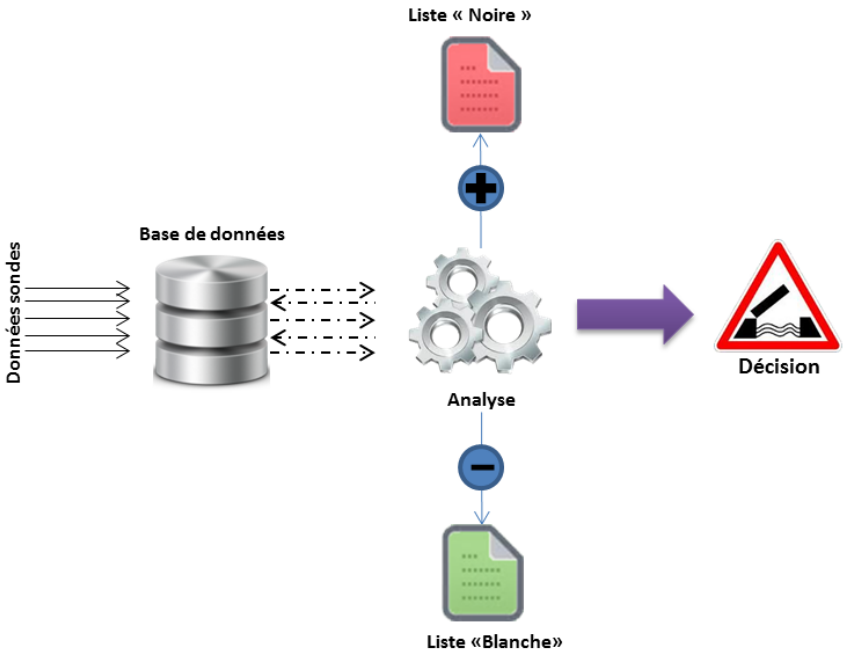

FIGURE 4.4 – Sous-système d'analyse et de décision

La méthode de décision qui sera adoptée doit permettre d'avoir des résultats probants dans des délais homogènes avec le besoin de réactivité en temps réel tout en gardant un bon niveau d'exactitude de résultats. Ainsi, les méthodes de prédiction seraient les plus adaptées à notre contexte (méthodes bayésiennes, estimation par maximum de vraisemblance ... etc.) même si toute leur difficulté c'est de choisir les paramètres avec des valeurs adaptées au contexte de prévention d'intrusion.

Dans notre contexte, pour l'expérimentation de la plateforme sans entrer dans la complexité de la méthode de décision, nous avons opté pour une méthode de décision simple qui est la classification en utilisant l'algorithme multi-dimensionnel de K-means.

Pour améliorer le taux de faux positif[4] (autrement dit réduire ce taux) un mécanisme de liste blanche permet d'éviter de bloquer le trafic à partir d'une source légitime comme un serveur de production générant un trafic réseau important par moment (par exemple lors d'une sauvegarde de données par le réseau). Toute adresse IP dans cette liste blanche sera exclue des règles de blocage générées automatiquement.

Par ailleurs, pour améliorer le temps nécessaire pour décider s'il faut ajouter une règle ou pas, une liste « noire » d'adresses IP identifiées comme étant sources d'attaques « sûres » permet de générer une règle de filtrage systématiquement à la première connexion provenant d'une telle source. Cette liste noire est en générale construite dans le temps par une communauté d'éditeurs de logiciels de sécurité. Un exemple typique d'une telle liste est utilisé dans les logiciels Antispam[5]. Nous pouvons alors poser le diagramme de décision

4. Faux positif : test positif à tort, par opposition à « faux négatif » qui est un test négatif à tort.
5. Antispam : logiciel contre les emails non sollicités, utilisés pour la publicité, les attaques phishing..etc.

suivant :

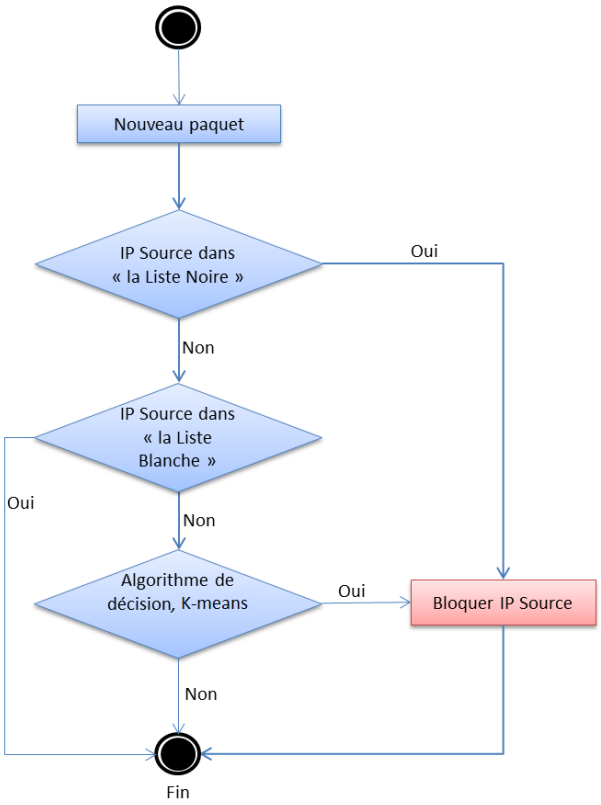

FIGURE 4.5 – Diagramme de prise de décision

4.2.2 Analyse de données : pre-traitement

Avant d'appliquer k-means sur les données il faut tout d'abord les comprendre et les traiter. Les données récupéré par les sondes contiennent plusieurs caractéristiques (date et temps de la capture du packet, l'identifiant du honeypot origine de la capture, l'identifiant de la source, le protocole de la transmission, type, port source, port destination, l'adresse IP source, code indicatif du pays, le nom du pays, la ville, l'abréviation de la ville, code postale, latitude et longitude). Nous pouvons remarquer que pas tous les champs sont utile pour notre analyses, et il y a des champs qui ne sont pas numérique. Pour cela nous avons fait un pre-traitement en deux étapes :

1. **Sélection de caractéristique :** Dans l'apprentissage machine et les statistiques, la sélection de caractéristiques, également appelée sélection de variables, sélection d'attributs ou sélection de sous-ensembles de variables, est le processus de sélection d'un sous-ensemble de caractéristiques pertinentes (variables, prédicteurs) à utiliser dans la construction du modèle. Les techniques de sélection des fonctionnalités sont utilisées pour quatre raisons :
   - simplification des modèles pour faciliter leur interprétation par les chercheurs / utilisateurs, [24]
   - temps d'apprentissage plus courts,
   - éviter le fléau de la dimension [6],

6. est un terme inventé par Richard Bellman en 1961 pour désigner divers phénomènes qui ont lieu lorsque l'on cherche à analyser ou organiser des données dans des espaces de grande dimension alors qu'ils n'ont pas lieu dans des espaces de dimension moindre.[50]

— généralisation améliorée en réduisant le sur-apprentissage [7] (formellement, réduction de la variance [24])

Le principe de base lors de l'utilisation d'une technique de sélection de caractéristiques est que les données contiennent des caractéristiques redondantes ou non pertinentes et peuvent donc être supprimées sans trop de perte d'informations [7]. Redondant et non pertinent sont deux notions distinctes, puisqu'un élément pertinent peut être redondant en présence d'un autre élément pertinent avec lequel il est fortement corrélé [18]. Nous avons sélectionné 8 caractéristiques parmi 15 : temps de la capture, l'identifiant du honeypot, identifiant de la source, le protocole, port source, port destination, code indicatif du pays et le nombre de connections successives de cette source. Ces attribues sélectionné améliorent la classification et les résultats de k-means.

2. **Encodage :** Nous remarquant que la plupart des caractéristique sélectionné dans l'étape précédente sont soit non numérique ou ils ont une très grande variance. Pour appliquer k-means nous avons besoin de représenter chaque entrée par un point dans un espace multi-dimensionnel, pour cela nous avons besoin d'encoder les attribues non numériques. Pour les caractéristiques qui ont une tres grandes variance nous avons choisis de les représenter en Cote Z. La cote Z correspond au nombre d'écarts types séparant un résultat de la moyenne. Elle se calcule de la même façon que la variable centrée réduite :

$$CoteZ = \frac{X - \mu}{\sigma}$$

où

— $CoteZ$: différence entre le résultat et la moyenne, divisé par l'écart-type

— X : valeur

— μ : Moyenne du groupe

— σ : Écart type du groupe

4.2.3 Analyse de données avec k-means

L'algorithme k-means est utilisé pour partitionner un ensemble d'observations donné en une quantité prédéfinie de k clusters. L'algorithme décrit par [31] commence par un ensemble aléatoire de k points-centrals (μ). À chaque étape de la mise à jour, toutes les observations x sont affectées au point central le plus proche (voir l'équation 4.1). Dans l'algorithme standard, une seule affectation à un centre est possible. Si plusieurs centres ont la même distance par rapport à l'observation, un centre aléatoire serait choisi.

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - \mu_i^{(t)} \right\|^2 \leq \left\| x_p - \mu_j^{(t)} \right\|^2 \ \forall\, j, 1 \leq j \leq k \right\} \tag{4.1}$$

Ensuite, les points centraux sont repositionnés en calculant la moyenne des observations attribuées aux points centraux respectifs (voir 4.2).

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{4.2}$$

Le processus de mise à jour se répète jusqu'à ce que toutes les observations restent aux points centraux assignés et, par conséquent, les points centraux ne seront plus mis à jour.

Cela signifie que l'algorithme k-means tente d'optimiser la fonction objectif 4.3. Comme il n'y a qu'un nombre fini d'assignations possibles pour le nombre de centroïdes et d'observations disponibles et que chaque itération doit aboutir à une meilleure solution, l'algorithme se termine toujours par un minimum local.

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2 \tag{4.3}$$

$$\text{avec} r_{nk} = \begin{cases} 1 & x_n \in S_k \\ 0 & \text{sinon} \end{cases}$$

Le principal problème de k-means est sa dépendance aux centroïdes initialement choisis. Les centroïdes pourraient finir par diviser des points de données communs, tandis que d'autres, séparés, seraient regroupés si certains des centroïdes étaient davantage attirés par des valeurs aberrantes. Ces points seront attirés vers le même groupe de points de données.

L'approche la plus courante consiste à effectuer plusieurs clusterings avec différentes positions de départ. Ensuite, le clustering qui a eu lieu le plus souvent est considéré comme correct. Une autre approche, plus récente, est la méthode dite de k-means++ d'Arthur et Vassilvitskii [2]. Cette extension de l'algorithme k-means essaie de répartir les centroïdes initiales sur les données afin de minimiser la probabilité de mauvais résultats. Les premiers points sont définis selon les auteurs par les étapes suivantes :

1. Prendre uniformément un point de données aléatoire à partir des données X et le marquer comme centroïde c_1
2. Choisissez un autre centroïde c_i avec la probabilité $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$ où $D(x)$ indique la distance la plus courte entre le point de données x et son centroïde le plus proche, déjà choisi.
3. Répétez 2. jusqu'à ce que tous les centroïdes initiaux soient choisis.

Ensuite, l'algorithme k-means standard décrit ci-dessus est exécuté. Les auteurs ont également montré qu'avec cet algorithme d'initialisation, k-means++ approximativement peut être calculé dans$O(\log n)$, par rapport à $O(n^{dk+1} \log n)$ pour l'algorithme standard.

Dans le cas notre expérimentation, nous avons implémenté k-means sous java. Nous avons choisis $k = 2$ qui représentent les 2 classes de données : dangereuses qui appartiennent dans la liste noire et les donnés non dangereuse qui appartiennent à la liste blanche. Comme nous avons des données qui sont représenté par 8 caractéristiques, nous avons utilisé la distance euclidienne dans l'espace n 4.4.

$$d(\mathbf{c}, \mathbf{x}) = d(\mathbf{x}, \mathbf{c}) = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \cdots + (x_n - c_n)^2} \tag{4.4}$$

Tel-que $n = 8$.

Pour différence entre les deux classes et les identifier comme dangereuse ou pas. Nous générons un trafic de données légitime à partir de notre réseau, et on calcule la distance entre ces données et les centroïde de chaque classe. La classe la plus proche de ces données est identifié comme classe de données non dangereuses.

4.3 Description du sous-système de distribution

Ce sous-système jour le rôle d'adaptateur entre le système de décision et le sous-système de filtrage. En effet, le sous-système de filtrage reste très varié et dépend des équipements réseau utilisés dans le SI 4.6. Ce module utilise des moyens adaptés selon le contexte pour configurer un sous-système de filtrage. Le module de contrôle est composé de l'ensemble des équipements et systèmes de l'entreprise qui offrent des mécanismes de filtrage. Ces solutions de contrôle peuvent être des routeurs, des commutateurs, des pares-feux, des machines avec un pare-feu . . . etc. Un telsystème doit être préalablement déclaré dans la base de données de l'IPMS pour qu'il soit exploitable et administrable à distance. Dans notre travail, nous prenons en charge les systèmes de contrôle supportant un accès par ligne de commande à distance, en particulier :

— Les routeurs : l'architecture prend en charge les routeurs (du constructeur Cisco en particulier) et est en mesure de modifier la configuration des listes de contrôle d'accès.

— Les pares-feux Netfilter : un système doté d'une installation de Netfilter [55] peut être pris en charge par l'architecture. Donc on peut ajouter des instructions IPTABLES sur une machine à distance; nous prendrons pour exemple les systèmes Linux.

L'ajout d'autres systèmes reste possible moyennant un développement spécifique dans le cas de systèmes propriétaires particuliers à un SI donné. Il suffira alors de le déclarer au niveau du sous-système de distribution.

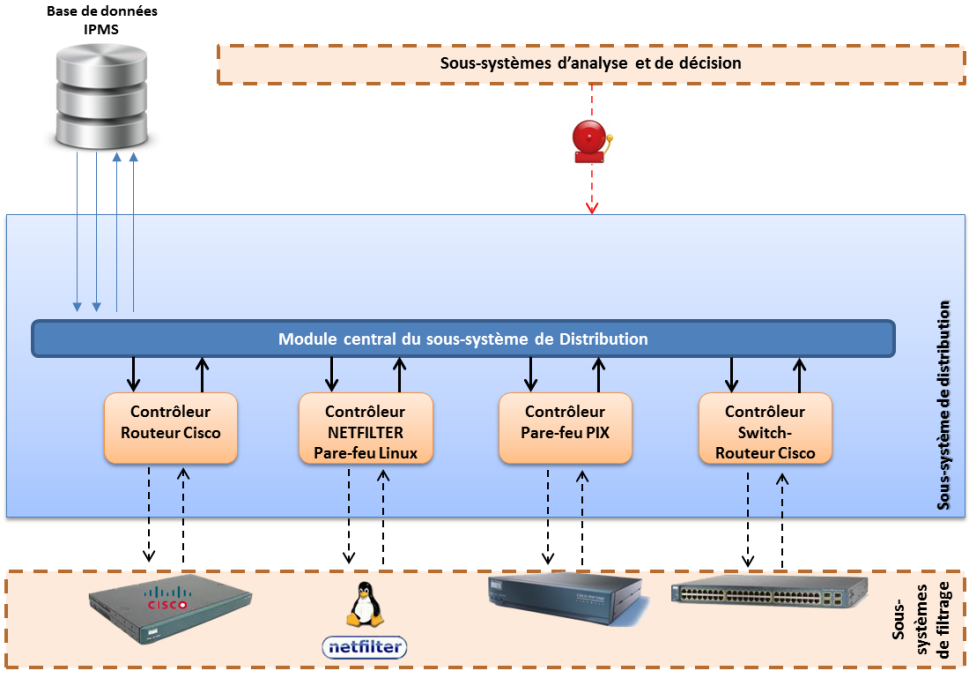

FIGURE 4.6 – Sous-système de distribution

4.4 Description du sous-système de gestion de l'IPMS

Le sous-système de gestion de l'IPMS est le module central permettant de gérer tous les autres sous-systèmes d'une manière centralisée, ceci inclus :

— La manipulation des sondes : ajout, modification et désactivation de sonde. Nous rappelons que le terme sonde renvoi à une instance d'installation de Honeyd,

— La manipulation des profils : par profil nous faisons référence aux différentes émulations possibles sur Honeyd. Donc une sonde peut avoir un ou plusieurs profils configurés.

— La manipulation des règles de filtrage : l'administrateur de l'IPMS peut intervenir sur les règles automatiquement générées par le sous-système d'analyse et de décision pour les désactiver ou les réactiver au grès de sa volonté La manipulation des règles de filtrage : l'administrateur de l'IPMS peut intervenir sur les règles automatiquement générées par le sous-système d'analyse et de décision pour les désactiver ou les réactiver au grès de sa volonté.

— La consultation et la gestion des logs remontés par les sondes : le nettoyage de la base de données des anciennes entrées estimées inutiles pour le fonctionnement de l'IPMS, mais aussi la sauvegarde desdits logs et leur historisation.

— La gestion des accès : ce n'est pas lié directement au fonctionnement de l'IPMS mais uniquement aux accès possibles à l'interface de gestion.

Nous pouvons schématiser ce système et tout l'IPMS comme présenté dans la figure 4.7 suivante :

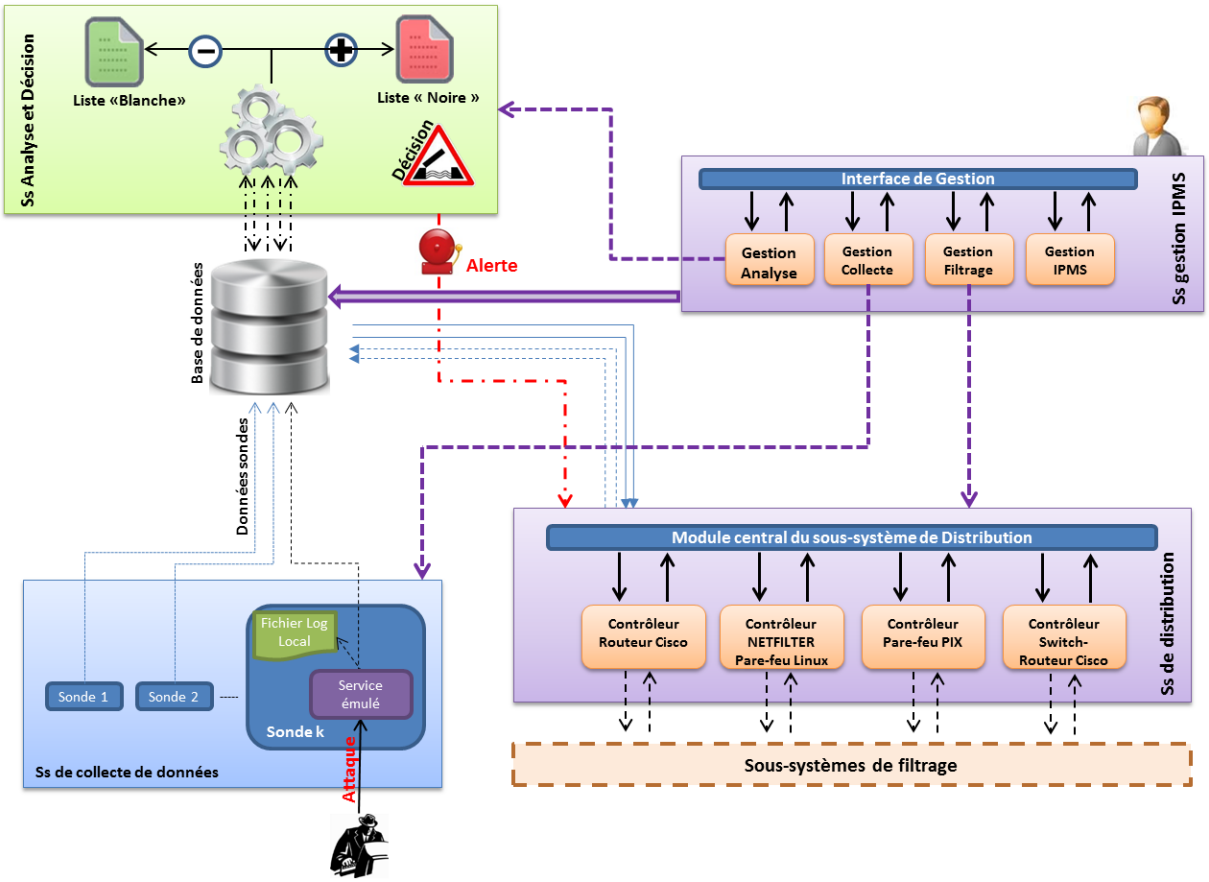

FIGURE 4.7 – Architecture de l'IPMS

le sous-système de gestion de l'IPMS est l'orchestre de toute la plateforme. En effet, à travers ce sous-système, nous devons être capables de réaliser toutes les actions d'administration de l'IPMS, par exemple :

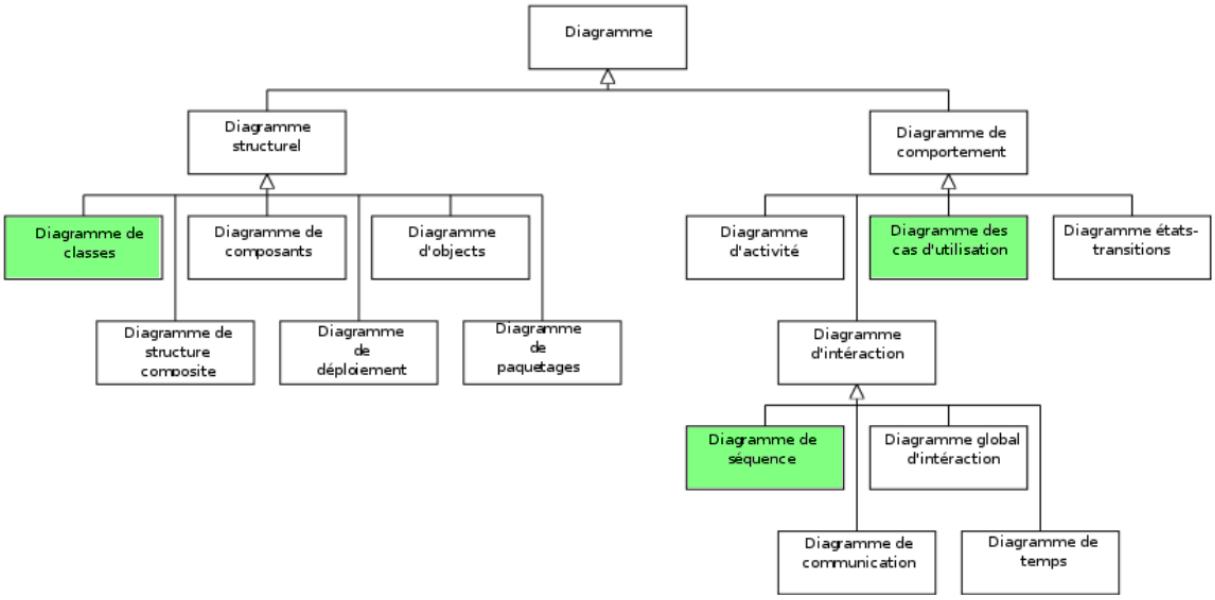

FIGURE 4.8 – Les 13 diagrammes définis dans UML 2.3

— Manipulation des sondes Honeyd, sous-système de collecte de données, y compris leur configuration et leur déploiement,

— Manipulation des règles de filtrage, qu'elles soient générées automatiquement par le sous-système de décision, ou ajoutées manuellement, ainsi que les listes blanche et noire.

— Consultation des différentes traces : traces des sondes, traces de toutes les modifications sur le système, ...etc.

La conception du sous-système de gestion a été faite à l'aide d'UML[7] [29]. Dans sa version 2.3, UML propose 13 diagrammes 4.8[13] dont le choix d'utilisation est laissé à l'appréciation de l'utilisateur selon son besoin. Lesdits diagrammes sont présentés ci-dessous à titre indicatif.

Pour ne pas alourdir ce paragraphe, on se limitera à présenter les diagrammes de cas d'utilisation, un des diagrammes de séquence et le diagramme des classes.

4.4.1 Conception

Nous présentons dans les paragraphes qui suivent les diagrammes UML retenus précédemment.

4.4.1.1 Diagrammes des cas d'utilisation

Les diagrammes de cas d'utilisation (Use Case) sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. De ce fait, nous allons présenter un diagramme par sous-système ainsi que le diagramme de tout l'IPMS.

7. UML : Unified Modeling Language, ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Il est couramment utilisé dans les projets logiciels, mais il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

Use Case général : l'IPMS Ce cas présente tous les sous-systèmes de l'IPMS ainsi que les interactions possibles 4.9. Nous avons principalement 3 acteurs, à savoir :

1. l'administrateur de l'IPMS, c'est lui le garant de son bon fonctionnement
2. l'attaquant : la source des ennuis,
3. l'équipement physique : c'est le sous-système de filtrage en général, donc un routeur, pare-feu ou n'importe quel équipement de filtrage

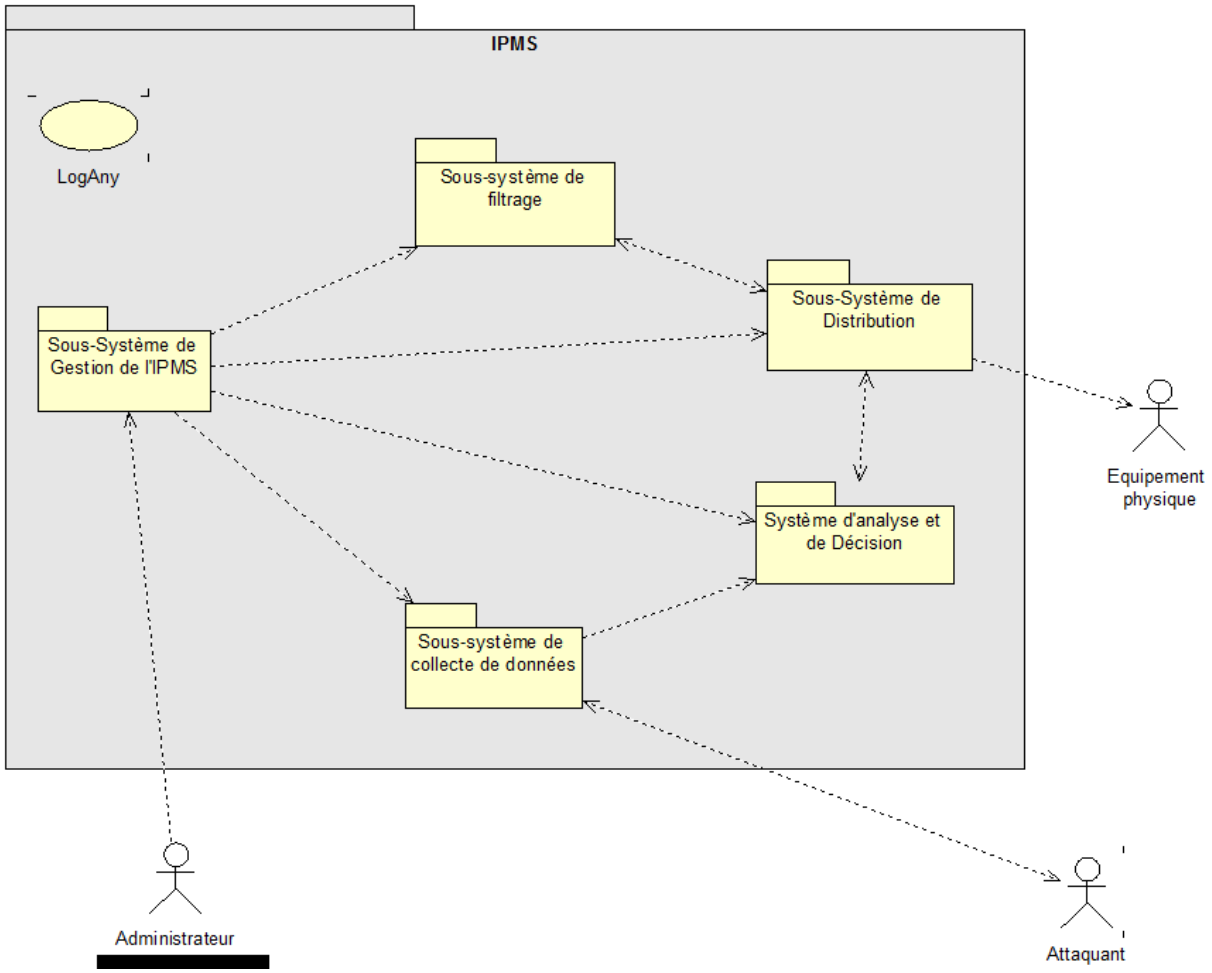

FIGURE 4.9 – Use Case général : IPMS

Use Case : Sous-système de gestion de l'IPMS Ce sous-système représente le backend de la plateforme IPMS. En effet il doit permettre l'administration de l'IPMS en tant qu'application. Autrement dit, il doit permettre de gérer la sauvegarde et la restauration de l'IPMS, le nettoyage de sa base de données et afficher les traces des activités sur le système. La figure suivante 4.10 représente ce diagramme.

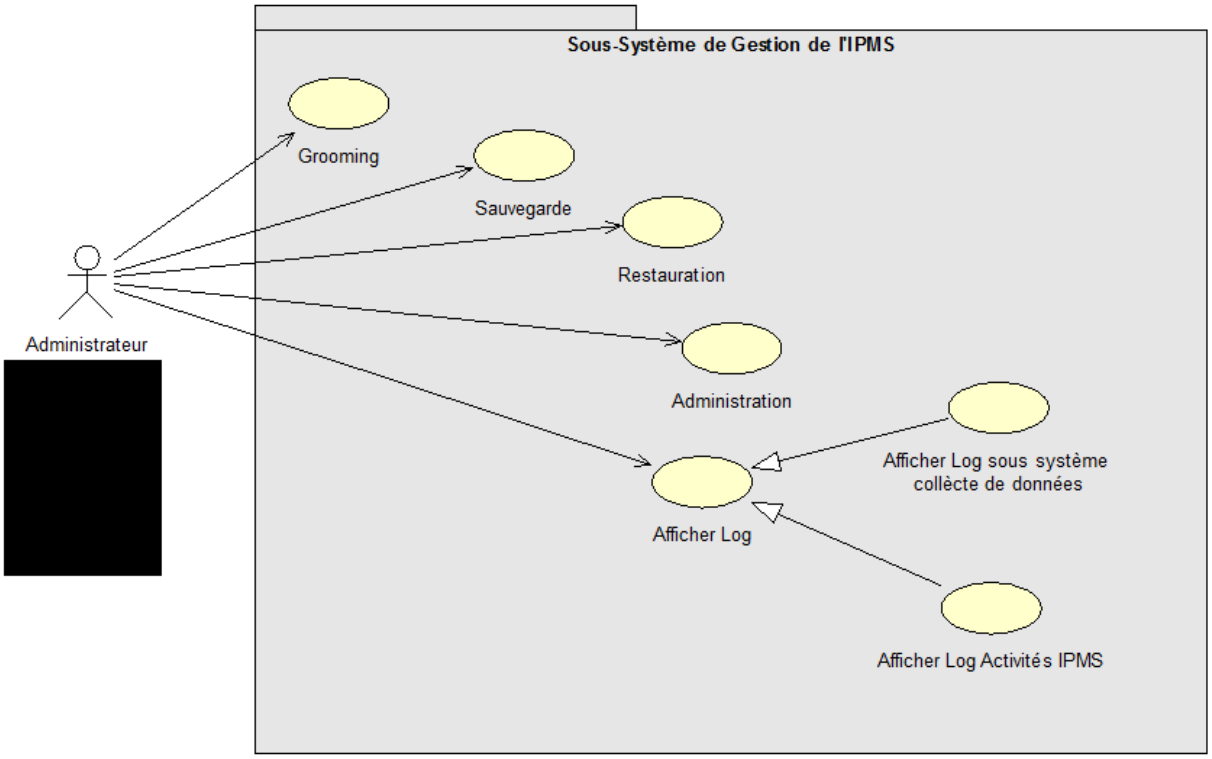

FIGURE 4.10 – Use Case du sous-système de gestion de l'IPMS

Use Case : Sous-système de collecte de données Concrètement le sous-système de collecte de données représente les sondes Honeyd. Donc ce diagramme de cas d'utilisation reflète les actions que l'administrateur fera dans le cadre de gestion des sondes, à savoir : ajout, modification et suppression d'une part, et configuration des profils Honeyd liés à une sonde donnée. Bien sûr, la sonde étant le point d'entrée d'une attaque, l'attaquant est aussi un acteur qui interagi avec la sonde et donc apparaîtra dans le diagramme 4.11.

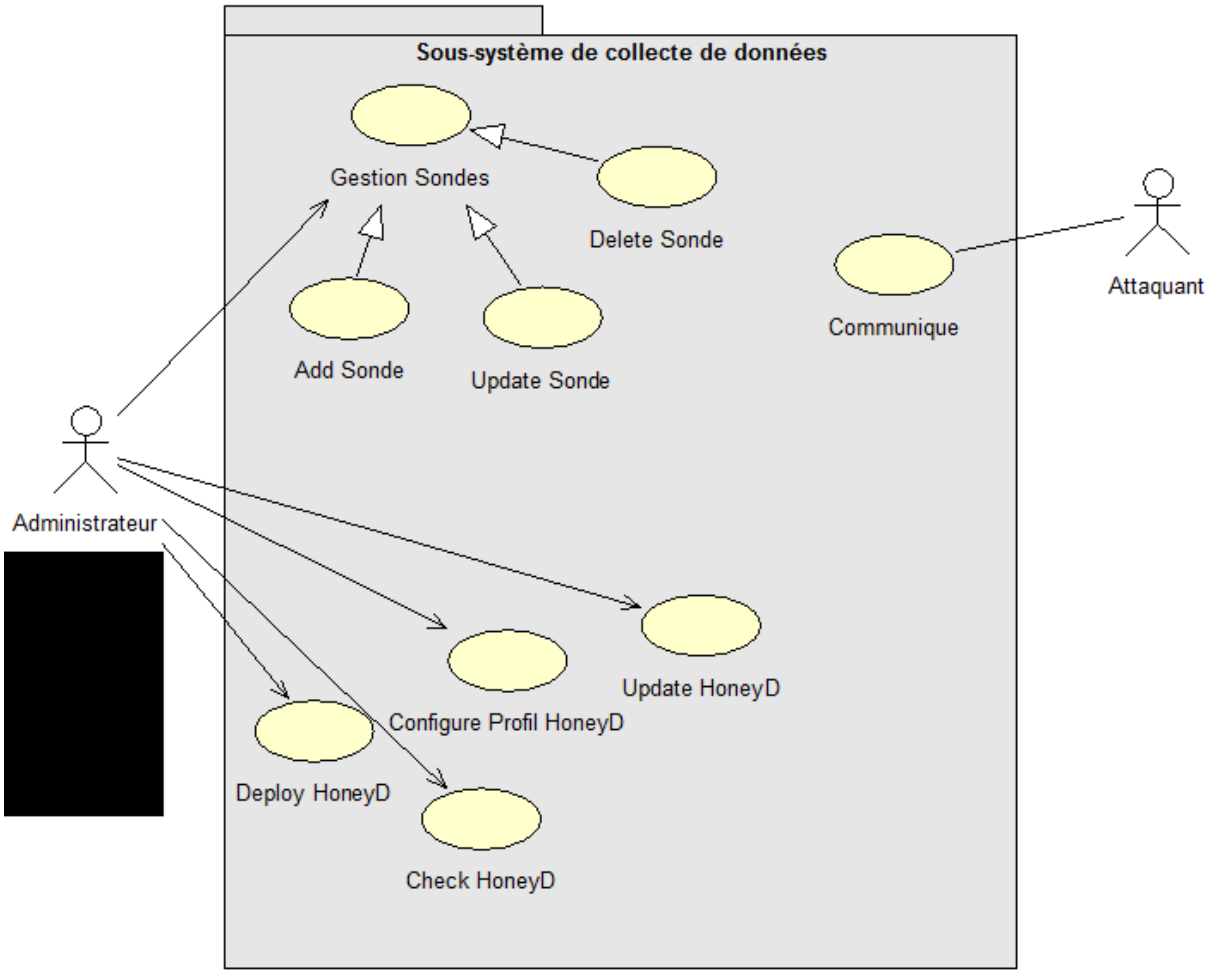

FIGURE 4.11 – Use Case du sous-système de collecte de données

Use Case : Sous-système de distribution Le sous-système de distribution interagi avec le sous-système de décision ainsi qu'avec celui de filtrage. Par ailleurs, l'administrateur interagi avec ce sous-système pour ajouter éventuellement une règle manuellement, ou pour désactiver une règle qui est devenue obsolète ou qui ne doit pas exister en général. D'où le diagramme représenté dans la figure suivante 4.12.

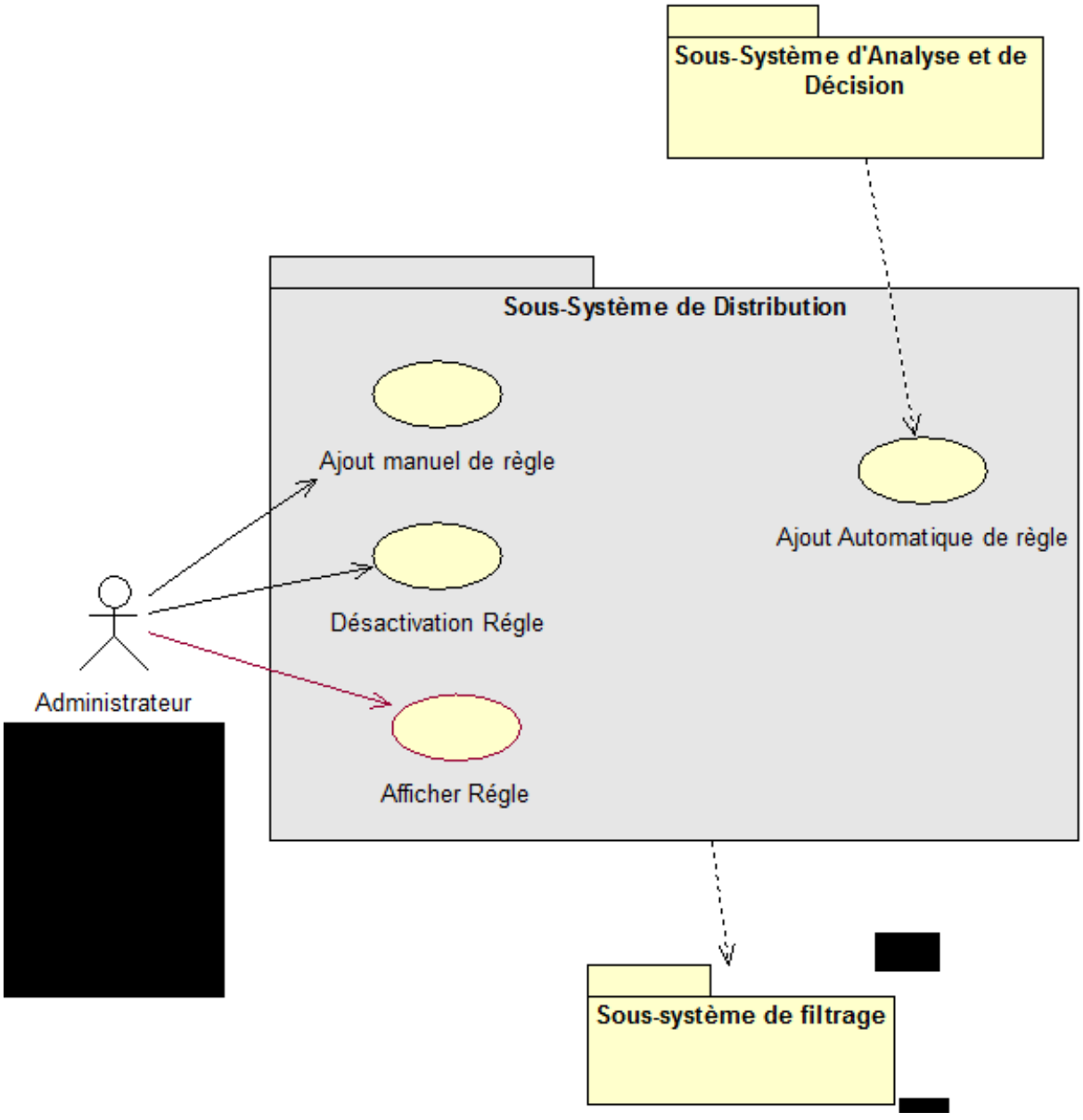

FIGURE 4.12 – Use Case du sous-système de distribution

Use Case : Sous-système de filtrage Le sous-système de filtrage représente concrètement le routeur ou le pare-feu qui sera configuré pour parer aux attaques. De ce fait, il faut pouvoir le configurer à distance à l'aide des scripts adéquats, et éventuellement de pouvoir visualiser sa configuration par l'administrateur de l'IPMS. Nous obtenons alors le diagramme de cas d'utilisations4.13 suivant :

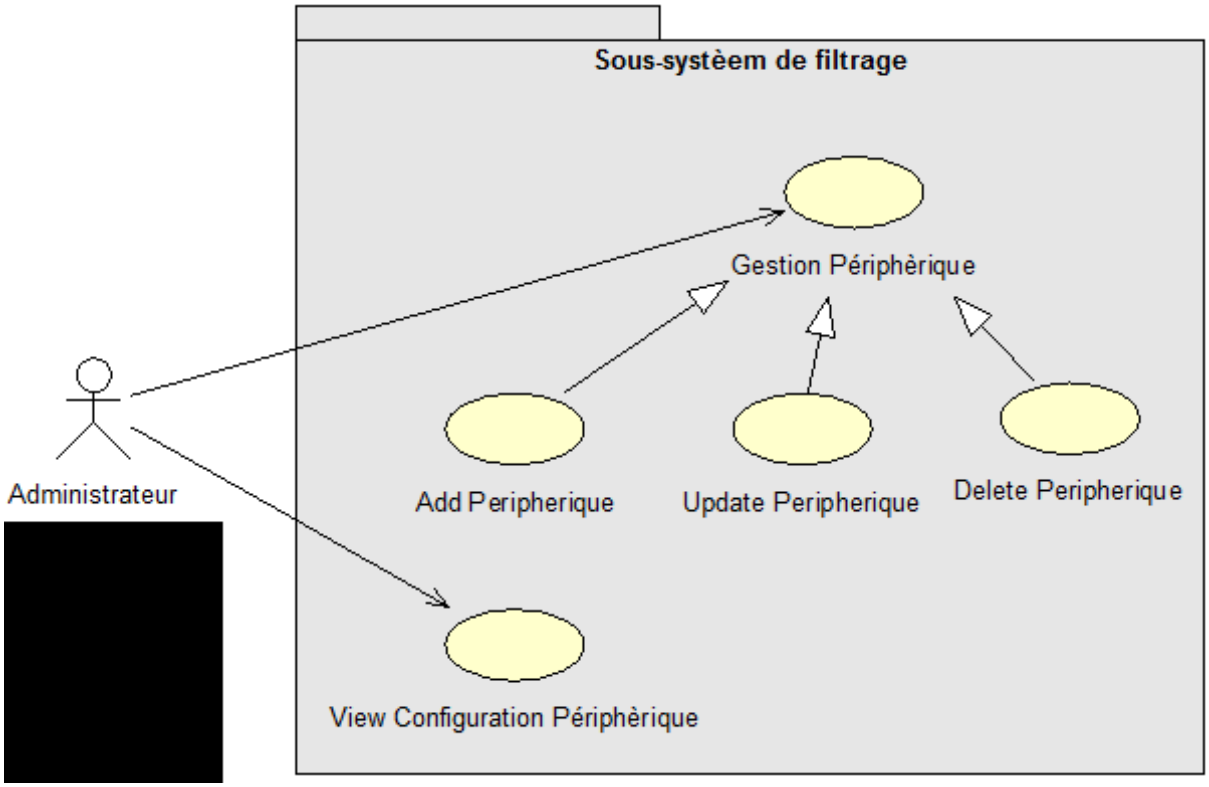

FIGURE 4.13 – Use Case du sous-système de filtrage

4.4.1.2 Diagrammes de Séquences

Les diagrammes de séquence trace les différents échanges de messages entre entités d'un système. Nous illustrons sur les trois figures 4.14, 4.15, 4.16 qui suivent les processus les plus importants de l'IPMS, à savoir : la collecte des données envoyées par l'attaquant, la génération d'une règle de filtrage et enfin le déploiement d'une telle règle.

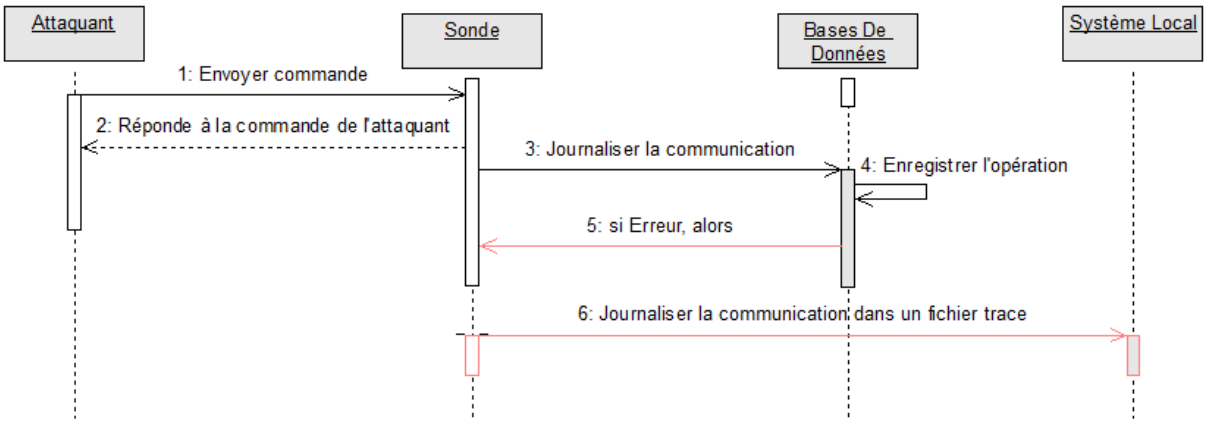

FIGURE 4.14 – Diagramme de séquence pour la journalisation de la communication entre attaquant et sonde

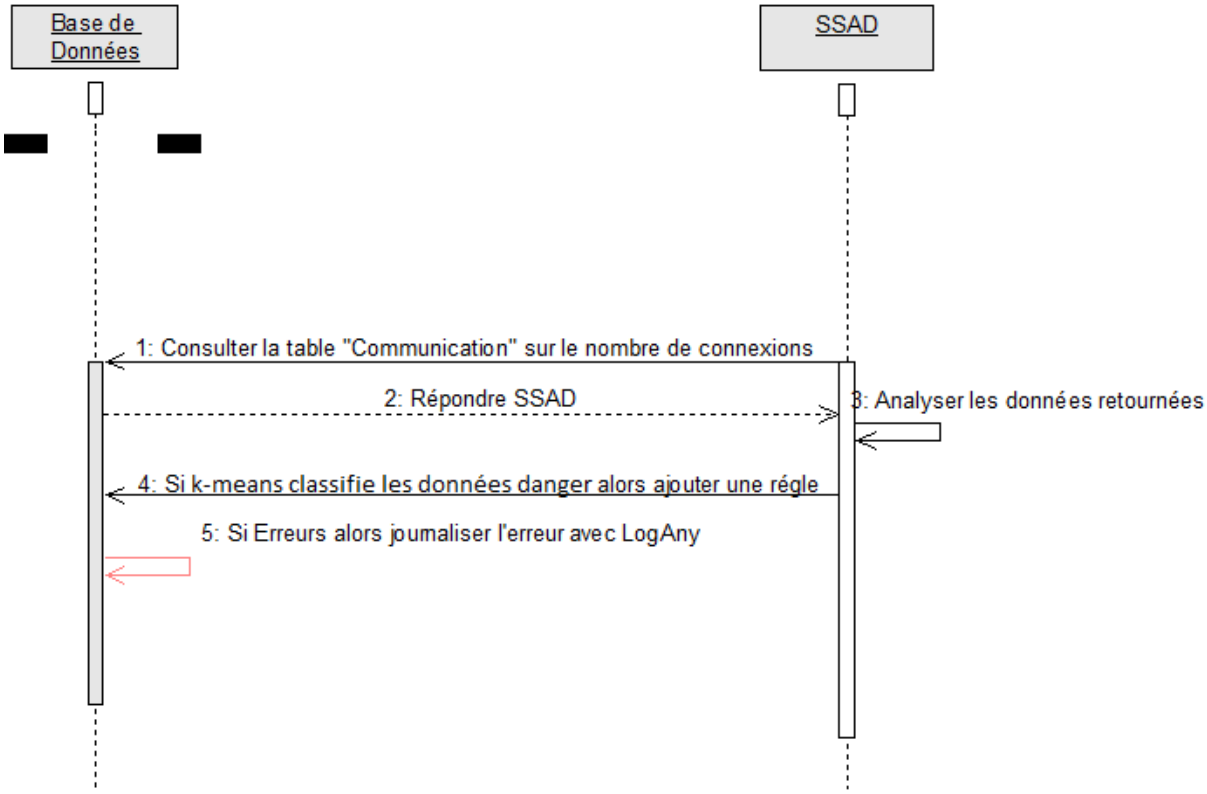

FIGURE 4.15 – Diagramme de séquence pour la génération de règle automatique

Dans le paragraphe qui suit nous présentons les diagrammes d'activité concernant ces trois diagrammes de séquences que nous avons présentés.

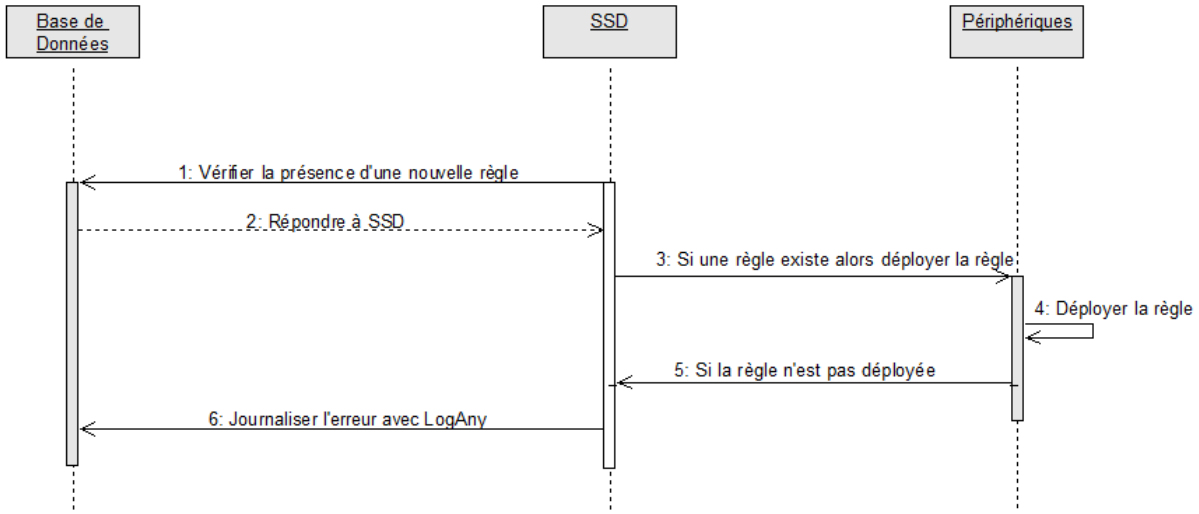

FIGURE 4.16 – Diagramme de séquence du déploiement d'une règle sur un périphérique

4.4.1.3 Diagrammes d'activité

Les diagrammes d'activité relatifs aux séquences que nous avons vus dans le paragraphe précédent sont présentés ci-dessous 4.17, 4.18, 4.19.

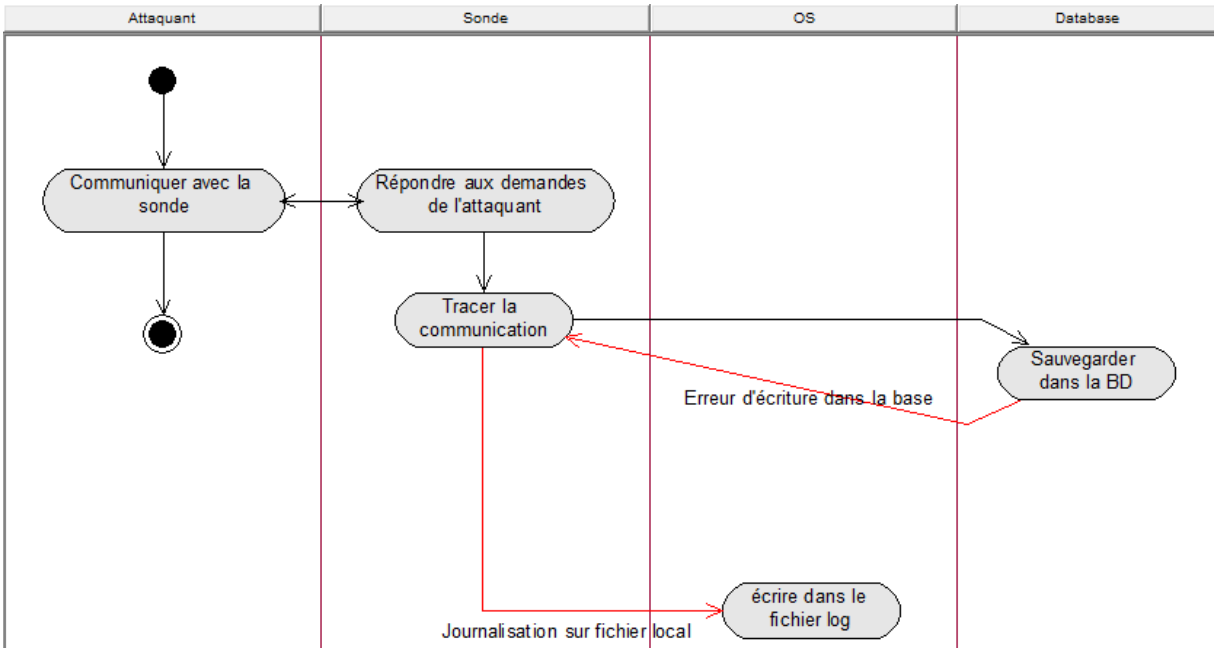

FIGURE 4.17 – Diagramme d'activité pour la journalisation des activités de l'attaquant

Dans le paragraphe qui suit nous présentons en dernier lieu le diagramme de classes de l'IPMS.

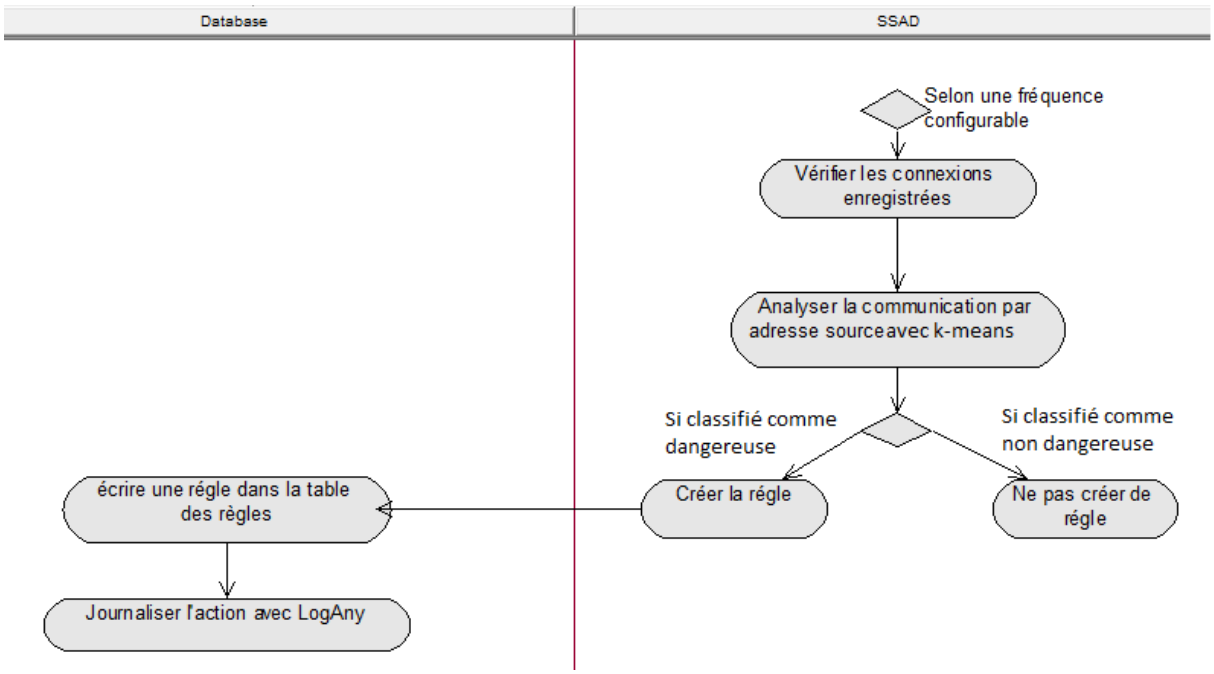

FIGURE 4.18 – Diagramme d'activité pour la génération automatique de règle

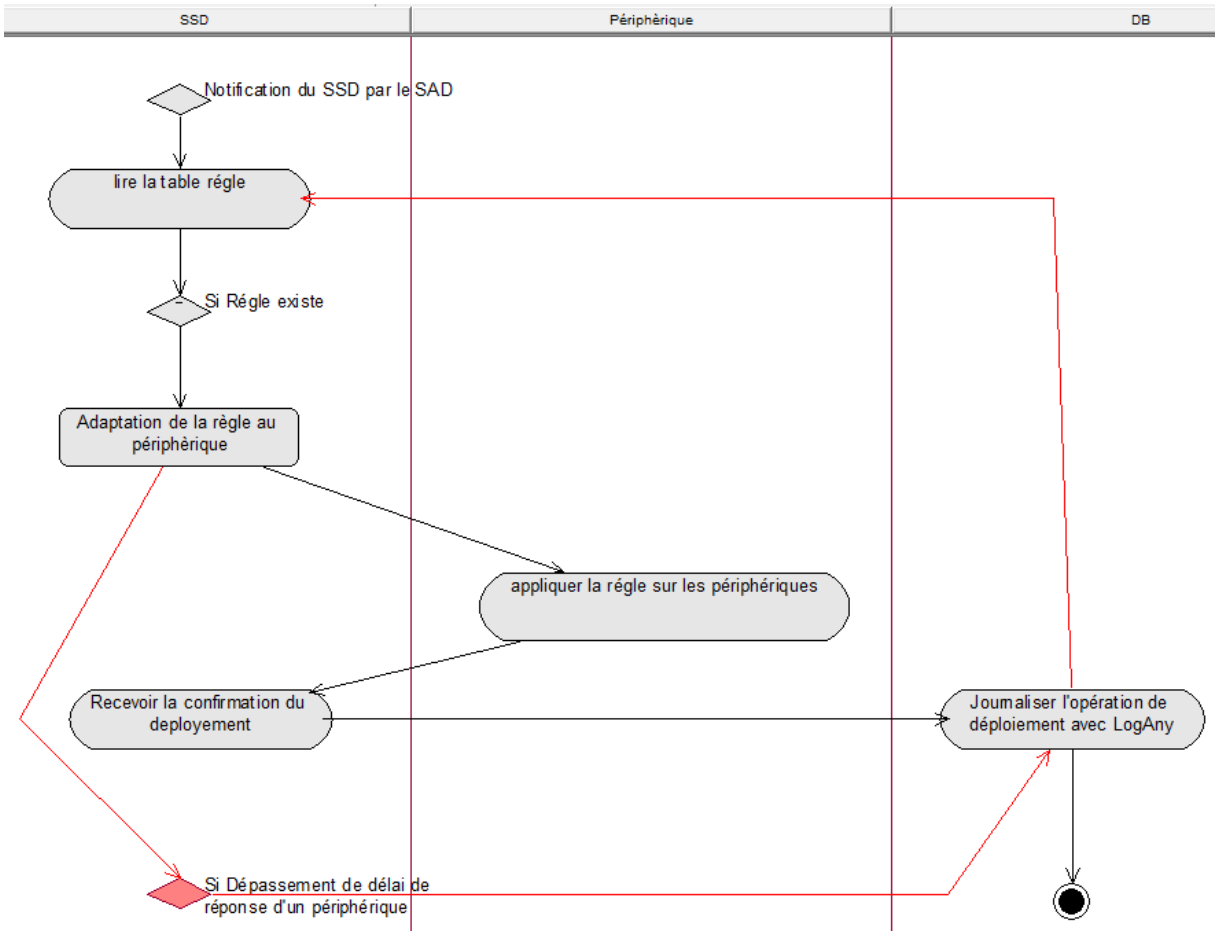

FIGURE 4.19 – Diagramme d'activité pour le déploiement automatique d'une règle

4.4.1.4 Diagrammes de classes

Le diagramme de classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques. Pour l'IPMS, on sort avec le diagramme de classes représenté sur la figure suivante :

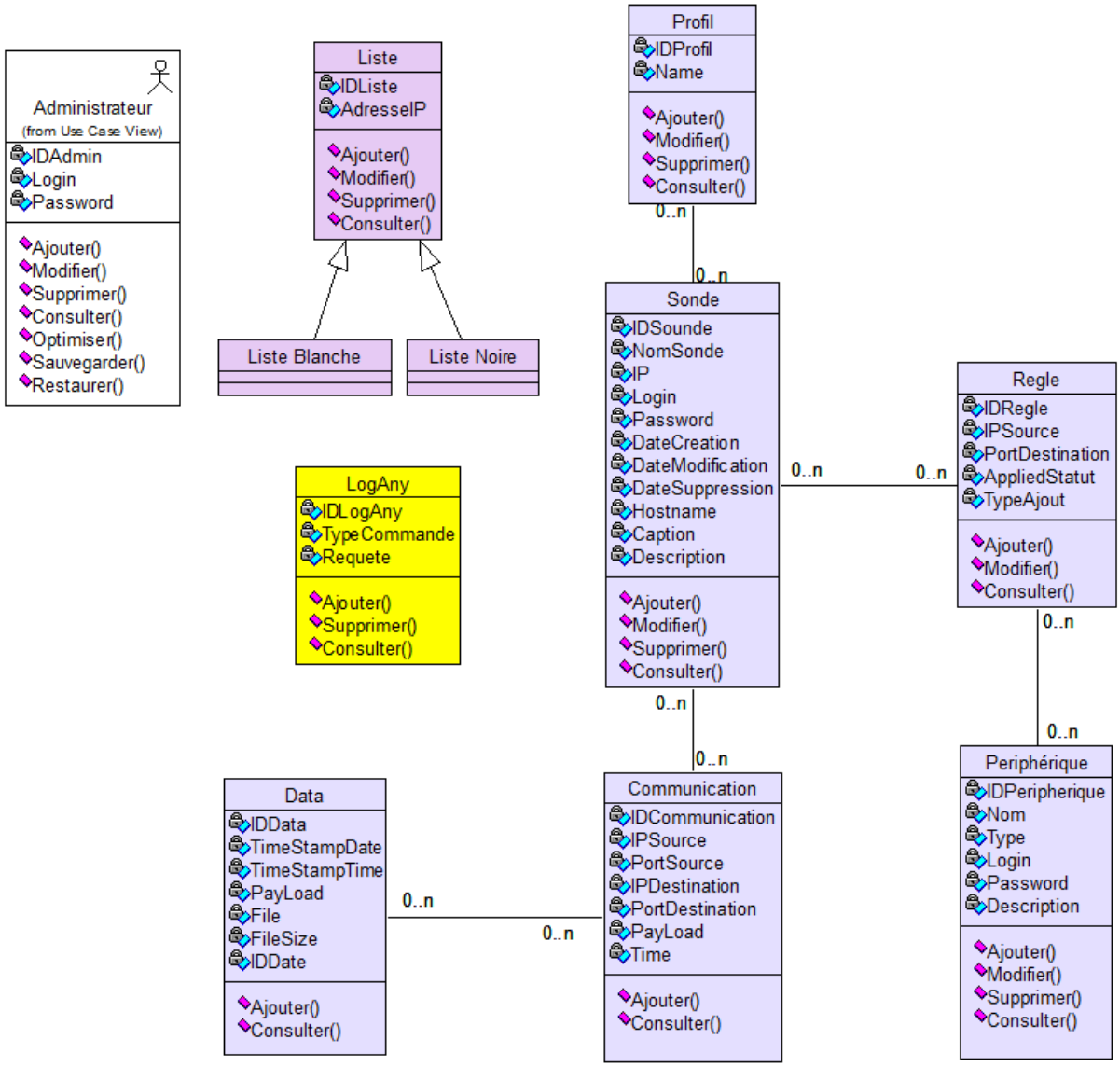

FIGURE 4.20 – Diagramme de Classes de l'IPMS

On distingue quatre grands blocs dans le diagramme de classes :

1. Classe « Administrateur » : c'est la classe nécessaire pour la gestion des comptes utilisateurs de l'IPMS. Elle offre aussi les méthodes pour la gestion de la base de données en termes de sauvegarde, restauration et optimisation.
2. Classe « LogAny » : c'est la classe relative à la fonctionnalité de journalisation de toute action sur l'IPMS, qu'elle soit faite par un administrateur ou automatiquement.

3. Classe « Liste » : c'est la classe pour la gestion des listes des adresses IP. Nous avons un héritage par liste, blanche et noir.
4. Les classes relatives à la gestion des sondes et aux données qui leurs sont relatives.
   (a) Classe « Sonde » : c'est la classe qui constitue le noyau de l'IPMS. Elle représente les sondes, ou sous-système de collecte de données.
   (b) Classe « profil » : c'est un script ou service simulé par une sonde. Une sonde peut simuler plusieurs services, ceci étant laissé au libre choix de l'administrateur de l'IPMS.
   (c) Classe « communication » : c'est la classe qui représente les connexions établies par un attaquant ou le retour qu'une sonde a fait sur une requête, sans pour autant stocker le contenu du paquet, ce dernier étant relatif à la classe « data »
   (d) Classe « Data » : cette classe représente le contenu de chaque paquet, ainsi que le chemin vers un fichier qui aurait été téléchargé vers la sonde par l'attaquant. Le fichier quant à lui est stocké dans le système de fichiers de la sonde.
   (e) Classe « règle » : comme son nom l'indique cette classe représente les règles qui seront déployées sur les périphériques, ou sous-systèmes de filtrage, qu'elles soient générées automatiquement ou renseignées manuellement par le gestionnaire de l'IPMS.
   (f) Classe « périphérique » : elle regroupe les informations nécessaires à la gestion d'un équipement de filtrage.

Ainsi, nous avons présenté les grands maillons de l'architecture de l'IPMS, ainsi que la conception UML de tous les sous-systèmes de cette solution de prévention d'intrusion. Nous discutons dans le paragraphe qui suit les résultats obtenu de l'expérimentation.

4.5 Résultat obtenu

La partie expérimentale du projet de ce mémoire est divisé en trois parties :

1. Conception et configuration réseau : cette partie consiste à
   — Créer réseau sur un l'émulateur GNS3[53] et configurer ses composants (serveurs, routeurs Cisco, firewalls Cisco PIX; ASA et honeypots) zone interne, DMZ et internet (externe),
   — installer les OS sur machine virtuel et configurer leurs accés réseau,
   — installer et configurer les honeypots HoneyD[42] sur les différentes zones.

   Après la première étape nous avons réussit à créer la topologie montré dans la figure 4.21.
2. Programmation du sous-système d'analyse et décision : dans cette partie nous avons programmé
   — un client java qui est déployé avec chaque Honeypot, qui envoie les fichier de capture périodiquement au serveur.
   — un serveur java qui as plusieurs rôles :
      — Recevoir les fichier log envoyé par les applications clients,

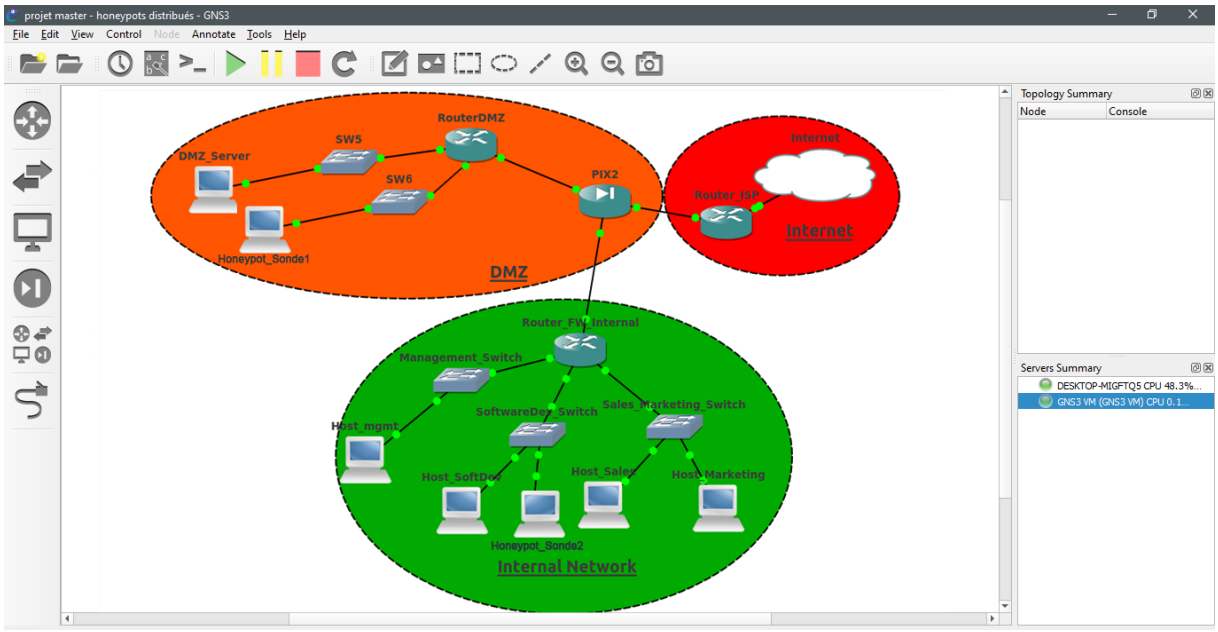

FIGURE 4.21 – Topologie réseau expérimentale

— lancer le calcule de classes avec k-means et entraîner le modèle,

— identifier une donnée comme dangereuse ou pas.

3. Tester le système d'analyse et de décision avec la data-set marx-geo [23]. Cette data-set contiens 451582 entrée, capturé par 9 honeypots répartis dans différents zones du réseau, pondant une durée de six mois (entre mars et septembre 2013). Nous avons ajouté à la data-set un trafic de donné légitime de la taille de 100000 capture pour que ça nous permet de calculer la précision de notre système. apres phase de pre-traitement où les données sont encodé et après sélection de caractéristique, nous avons répartis les donnés en deux parties (entraînement et teste) ou nous avons varié le pourcentage des données utilisés pour faire l'apprentissage de notre modèle, et nous avons obtenu les résultats présenté dans la figure 4.22.

4.6 Positionnement de la solution proposée

La mise en place de cette plateforme a privilégié en premier lieu le « zéro » impact sur le réseau à protéger. En effet, la mise en place de cette solution peut se faire en parallèle avec tout réseau de production sans impact sur le rendement de l'infrastructure existante. Selon le degré d'investissement consenti pour la protection de l'infrastructure de l'entreprise, cette dernière peut opter pour la mise en place d'un réseau parallèle pour la communication entre sondes et serveur central hébergeant la base de données, ou bien utiliser le réseau existant avec des communications chiffrées sans impact majeur. En effet, la taille des données échangées entre sondes et serveur central n'est pas trop importante pour ralentir le réseau local qui, en général, a un débit de 100Mbps sinon 1Gbps pour les grandes structures. Un autre point non moins important est l'aspect distribué et évolutif (dans le sens Scalability) de la plateforme. En effet, nous pouvons ajouter des sondes partout dans le réseau de l'entreprise, y compris dans au niveau filiales distantes sans avoir à dupliquer la plateforme d'analyse

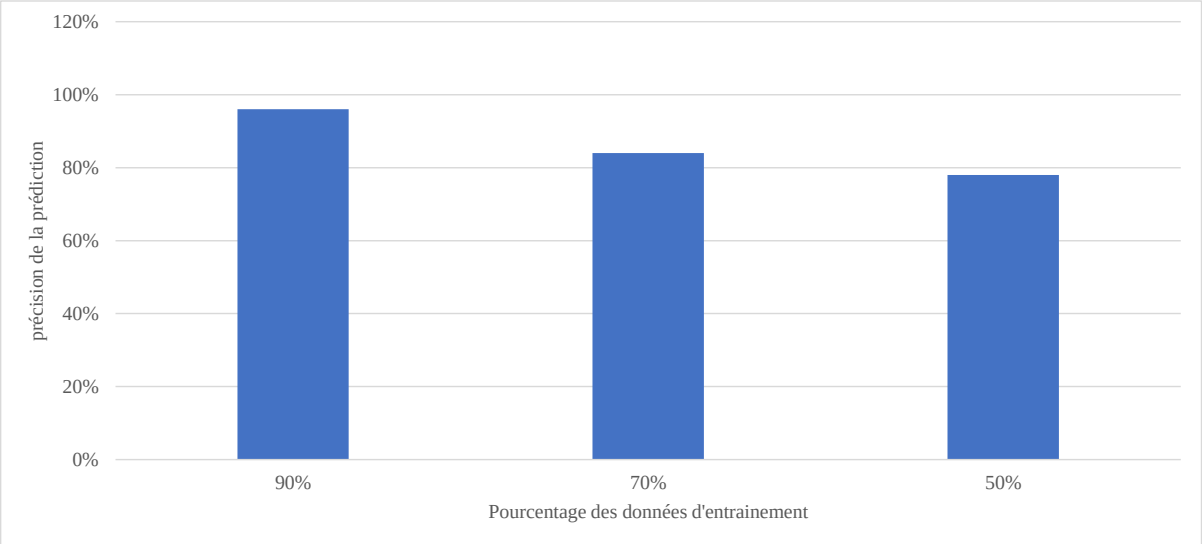

FIGURE 4.22 – La précision de la prédiction par rapport au pourcentage de données d'apprentissage

et de décision. Ceci est d'encore plus simple que Honeyd offre la possibilité de créer des tunnels GRE [8] virtuels. Par ailleurs, l'ajout d'une sonde ne nécessite aucune reconfiguration de la plateforme ni un cout élevé puisqu'il s'agit seulement d'un paramétrage standard de la sonde. Nous pouvons reprendre le tableau comparatif des IPS classique et nouvelle génération pour avoir une idée sur le positionnement de la solution IPMS.

Fonctionnalité clé	IPS Classique	NGIPS	IPMS
Modes IPS Inline et IDS passif	✓	✓	
Politique de détection par défaut	✓	✓	✓
Rapports, Alertes et tableau de bord	✓	✓	✓
Règles personnalisées		✓	✓
Protection basée sur les vulnérabilités		✓	✓
Analyse automatique d'impact		✓	✓
Optimisation automatique		✓	✓
Traçabilité d'identité des utilisateurs		✓	
Surveillance d'applications		✓	
Analyse comportementale du réseau		✓	
IPS virtuel et console de gestion		✓	✓

TABLE 4.1 – Positionnement de l'IPMS par rapport aux IPS classiques et NG

Avec un score de 7/11 nous pouvons dire que la plateforme IPMS s'approche plus des IPS Nouvelle Génération que des IPS classiques.

8. GRE : Generic Routing Encapsulation, c'est un protocole de tunneling développé par Cisco Systems qui permet d'encapsuler une grande variété de protocoles de couche réseau dans des liaisons point à point à travers Internet.

4.7 Conclusion

Après l'analyse de nos résultat nous pouvons remarquer que notre système peux arriver à un tôt de précision très élevé mais il a besoin d'une phase d'entraînement très longue, qui peux durer plusieurs mois. La raison de cette longue durée est la complexité de l'espace qui caractérise les données (8 dimensions). Malgré la bonne conception de tout les sous-systèmes du IPMS, la programmation et l'évaluation comprenais seulement les sous-systèmes de collecte de données et le sous-système d'analyse et de décision, à cause de la durée limité du projet de mémoire.

Conclusion générale

4.8 Introduction

De plus en plus d'entreprises s'ouvrent sur le monde en exposant leurs produits et services sur Internet. Cette ouverture concerne aussi un partage d'informations avec les partenaires avec des accès privilégiés d'une part et d'autre aux systèmes d'informations qui englobent toutes les informations métier mais aussi business de chacun des deux partenaires (ou plus). Cette ouverture, aussi minime qu'elle soit, offre des portes d'entrées, visibles et dérobées, à tout genre d'utilisateurs. Ceci implique un grand risque pris surtout en voyant la nouvelle guerre cybernétique qui n'épargne ni organismes gouvernementaux ni bancaires ni même le plus petit des sites e-commerce. En effet, il y a un marché important pour les informations confidentielles que pourrait récupérer un hacker avisé et bien outillé. A titre d'exemple, nous entendons ces jours ci que la boutique en ligne du gigantesque Apple a été cible d'attaque qui a fait que toutes les informations bancaires des clients d'Apple sont entre les mains et à la merci d'un grand réseau de piratage informatique. Bien sûr, les entreprises ont pris conscience de ces dangers et dépensent chaque année des sommes colossales d'argent pour s'équiper en équipements de filtrage plus robustes, en équipement de détection d'intrusion, et pour recruter des profils spécialisés dans la sécurité informatique. Ce jeu malsain entre hackers et éditeurs de solutions de sécurité donne souvent raison aux hackers qui innovent chaque jour pour contourner les mesures de sécurité en place. Ceci a donné lieu à une nouvelle gamme d'outils de sécurité appelés outils de prévention d'intrusion. Ces outils sont plus intelligents et moins figés dans leurs méthodes de décision pour pouvoir déceler le moindre flux réseau suspect et désamorcer une bombe logique avant qu'elle n'explose.

4.9 Apport du mémoire

Dans le même esprit, nous avons essayé dans ce travail de mettre en place un outil de prévention d'intrusion qui soit adapté à l'envergure des réseaux informatiques actuels. En effet, avec son architecture distribuée il est approprié pour les grands réseaux d'entreprises, à faible impact sur le réseau de production et évolutif par nature. Pour ce faire, nous nous sommes basés sur les pots de miel, un outil qui offre une grande richesse par rapport aux données qu'il peut collecter. Ces données sont la base sur laquelle le système d'analyse et de décision décide si un trafic est normal ou bien suspect. L'aspect distribué fait en sorte que les possibilités de collecte sont plus grandes et peuvent aussi couvrir le réseau interne de l'entreprise. Ces derniers ont la réputation d'être sûrs juste du fait qu'ils sont dans quelques cas isolés d'Internet. Cette affirmation est d'autant plus

fausse qu'une étude du CERT [57] a montré que presque les trois quarts des attaques que subissent les entreprises proviennent de l'intérieur. Parmi les raisons de cette situation : la mobilité des utilisateurs. En effet, la majorité des utilisateurs accèdent à des sites web *de confiance*, et reçoivent des emails de publicité de site *de confiance*. Ce leurre de confiance fait qu'il y a des *malwares* qui circulent et à eux seuls ils représentent 38% des attaques internes aux entreprises. 48% de ces dernières sont dues à une élévation de privilèges à cause d'une mauvaise configuration de droits alors que le hacking ne représente que 40% [9] du lot. [21]

Nous avons mis en place une architecture qui permet de renforcer le niveau de sécurité vu qu'elle permet de bloquer les attaques dès leurs premiers signes et ce sans intervention humaine. La liste blanche utilisée dans la partie analyse permet d'éviter qu'un blocage de la production ait lieu et donc améliorer le taux de faux positif qui est le principal point à risque pour tout IPS peu importe sa nature. L'IPMS permet de répondre aux besoins exprimés par les administrateurs de sécurité, les intégrateurs des plateformes de sécurité ainsi que les organisations soucieuses de la sécurité de leurs environnements. Dans la plus part des cas les besoins suivants reviennent :

1. Palier aux faiblesses des IPS réseau standards, à savoir :
   (a) corriger l'aspect « maillon faible »
   (b) être invisible dans le réseau du SI
   (c) réduire le taux de faux positifs qui nuisent au bon fonctionnement global d'un réseau informatique.
2. Avoir un système réparti dans tous les segments d'un réseau de SI
3. Ne plus avoir de goulet d'étranglement dans le réseau (cas des IPS dits inline)
4. Avoir un IPS qui s'intègre dans un réseau avec zéro impact sur l'existant.
5. Avoir un IPS avec un système de décision adaptable à la nature du métier sans changer toutes les composantes d'un SI.
6. Avoir un point de gestion central qui :
   (a) Donne une vision globale sur l'état de sécurité et les risques encourus par un système d'information.
   (b) Donne la possibilité de gérer les différents équipements de filtrage d'une manière centralisée,
   (c) Donne la possibilité d'ajouter des sondes de collecte de données et de les configurer automatiquement.

4.10 Perspectives

Certes cette solution répond à plusieurs besoins et répond à plusieurs critères des IPS nouvelles génération, mais il reste plusieurs aspects et axes d'amélioration.

— La méthode de décision utilisée k-means, prend beaucoup de temps pour l'entraîner et la déployer, il est important d'implémenter un autre algorithme qui soit plus évolutif et qui s'adapte au contexte de

9. Les pourcentages ne sont pas cumulatifs mais représentent à chaque fois l'implication de la valeur sur le nombre total d'incidents.

l'entreprise. Les méthodes d'apprentissage machine se développent de plus en plus, autres algorithme et méthode d'intelligence artificiel peuvent ètre testé.

— Le choix du pot de miel à utiliser comme sonde. Nous avons opté pour Honeyd, mais il y a d'autres pots de miel qui peuvent être testés. Des pots de miels sont encore en cours de développement et profiter de cet aspect pour couvrir le maximum de failles de sécurité rendra la plateforme encore plus efficace. Par ailleurs, les derniers pots de miel supportent le standard IDMEF [10] [16] ce qui permettra l'échange de messages avec les outils de détection d'intrusion du marché. Ceci améliorera la collaboration entre équipements de sécurité du réseau pour la prévention d'intrusion.

— Ajouter un coefficient de pondération de la criticité selon l'emplacement de la sonde. Si une attaque arrive jusqu'à une sonde placée dans le même réseau que les serveurs de données d'autres serveurs critiques ça voudrait dire un très grand risque de défaillance en sécurité qu'une sonde placée dans la DMZ.

— Implémenter les autres sous-systèmes du système, et améliorer l'interface de gestion pour implémenter un module de notification par SMS dans le cas où un critère de criticité est vérifié.

— Intégrer un module d'analyse proactive et automatique des données et statistiques qui aidera à améliorer les règles et l'algorithme de décision.

10. IDMEF : Intrusion Detection Message Exchange Format

Bibliographie

[1] Hariu Takeo Akiyama Mitsuaki, Kawakoya Yuhe. Scalable and performance-efficient client honeypot on high interaction system. In *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet*, pages 40–50. IEEE, 2012.

[2] David Arthur and Sergei Vassilvitskii. K-means++ : The Advantages of Careful Seeding. In *Proc. Eighteenth Annu. ACM-SIAM Symp. Discret. Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[3] Information Technology at Johns Hopkins. `http://www.it.jhmi.edu/glossary/ghi.html`.

[4] Pomsathit Auttapon. Effective of unicast and multicast ip address attack over intrusion detection system with honeypot. In *2012 spring congress on engineering and technology*, pages 1–4. IEEE, 2012.

[5] ThU Aye Aye. Integrated intrusion detection and prevention system with honeypot on cloud computing environment. *International Journal of Computer Applications*, 67(4), 2013.

[6] Reiser Hans P Beham Michael, Vlad Marius. Intrusion detection and honeypots in nested virtualization environments. In *2013 43rd Annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pages 1–6. IEEE, 2013.

[7] Spiliopoulou Athina Hayward Caroline Rudan Igor Campbell Harry Wright Alan F Wilson James F Agakov Felix Navarro Pau others Bermingham Mairead L, Pong-Wong Ricardo. Application of high-dimensional feature selection : evaluation for genomic prediction in man. *Scientific reports*, 5 :10312, 2015.

[8] Ammar BOULAICHE. *TECHNOLOGIES HONEYPOTS*. Mémoire de magistère en informatique, Université Abderrahmene Mira de Béjaia, 2006.

[9] Neumann Jason C. *The book of GNS3 : build virtual network labs using Cisco, Juniper, and more*. No Starch Press, 2015.

[10] Patel Ankit D Chawda Kartik. Dynamic & hybrid honeypot model for scalable network monitoring. In *International conference on information communication and embedded systems (ICICES2014)*, pages 1–5. IEEE, 2014.

[11] Welsh Chris. *GNS3 network simulation guide*. Packt Publ., 2013.

[12] Yongbo Zhang Dongxia Liu. An intrusion detection system based on honeypot technology. In *2012 international conference on computer science and electronics engineering*, volume 1, pages 451–454. IEEE, 2012.

[13] Diagrammes définis dans UML v2.3. https://bit.ly/2WYkOxf.

[14] Chovancová E Fanfara P, Dufala M. Usage of proposed autonomous hybrid honeypot for distributed heterogeneous computer systems in education process. In *2013 IEEE 11th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 83–88. IEEE, 2013.

[15] Susan Fogarty. Gns3 network simulator raises its game. Network Computing. UBM Tech, 2015.

[16] Network Working Group. The intrusion detection message exchange format (idmef), rfc 4765. http://goo.gl/G8rfx.

[17] Liu Zhenxiang Guizhou Jiang Zhen. New honeypot system and its application in security of employment network. In *2012 IEEE symposium on robotics and applications (ISRA)*, pages 627–629. IEEE, 2012.

[18] Elisseeff André Guyon Isabelle. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar) :1157–1182, 2003.

[19] Bektaş O. Soysal M. Orcan S. Gökırmak Y., Yüce E. Ipv6 balküpü tasarımı. In *Tübitak Ulakbim*, Ankara, 2011.

[20] Soysal M. Yiğit S. Gökırmak Y., Bektaş O. Sanal ipv6 balküpü ağı altyapısı : Kovan. In *Ulusal IPv6 Konferansı*, 2011.

[21] Hackfest. L'intranet si sécuritaire. http://goo.gl/sNLs6.

[22] Hamou-Lhadj Abdelwahab Hassine Jameleddine. Toward a ucm-based approach for recovering system availability requirements from execution traces. In *International Conference on System Analysis and Modeling*, pages 48–63. Springer, 2014.

[23] Jay Jacobs. Inspecting internet traffic. https://bit.ly/2ZnltWV.

[24] Hastie Trevor. Tibshirani Robert James Gareth, Witten Daniela. *An introduction to statistical learning*, volume 112. Springer, 2013.

[25] Sardana Anjali Joshi RC. *Honeypots : A new paradigm to information security*. CRC Press, 2011.

[26] Watson Robert NM Kamp Poul-Henning. Jails : Confining the omnipotent root. In *Proceedings of the 2nd International SANE Conference*, volume 43, page 116, 2000.

[27] Nicopolitidis Petros Obaidat Mohammad Koniaris Ioannis, Papadimitriou Georgios. Honeypots deployment for the analysis and visualization of malware activity and malicious connections. In *2014 IEEE international conference on communications (ICC)*, pages 1819–1824. IEEE, 2014.

[28] Spitzner Lance. *Honeypots : tracking hackers*, volume 1. Addison-Wesley Reading, 2003.

[29] UML : Unified Modeling Language. Object management group (omg). https://www.uml.org/.

[30] Huang Wei Li Song, Zou Qian. A new type of intrusion prevention system. In *2014 international conference on information science, electronics and electrical engineering*, volume 1, pages 361–364. IEEE, 2014.

[31] J MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. Math. Stat. Probab. Vol. 1 Stat.*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[32] Phillip Maddux. Honeypy docs. `https://honeypy.readthedocs.io/en/latest/`.

[33] Mantrap. Recourse technologie. `www.recourse.com/products/mantrap/trap.html`.

[34] Massoth Michael Markert Jürgen. Honeypot effectiveness in different categories of attacks on wireless sensor networks. In *2014 25th international workshop on database and expert systems applications*, pages 331–335. IEEE, 2014.

[35] Bendriss El Mehdi. *Architecture distribuée pour la prévention d'intrusions basée sur les honeypots*. PhD thesis, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS), Rabat, 2014.

[36] Kwok Lam-For Meng Yuxin. Adaptive blacklist-based packet filter with a statistic-based approach in network intrusion detection. *Journal of Network and Computer Applications*, 39 :83–92, 2014.

[37] Deaconescu Razvan Musca Constantin, Mirica Emma. Detecting and analyzing zero-day attacks using honeypots. In *2013 19th international conference on control systems and computer science*, pages 543–548. IEEE, 2013.

[38] BACK OFFICER. `http://www.nfr.com/resource/backOfficer.php`.

[39] Mishra Bimal Kumar Paul Sounak. Honeypot based signature generation for defense against polymorphic worm attacks in networks. In *2013 3rd IEEE International Advance Computing Conference (IACC)*, pages 159–163. IEEE, 2013.

[40] Niyogi Rajdeep Pilli Emmanuel S, Joshi RC. A generic framework for network forensics. *International Journal of Computer Applications*, 1(11) :1–6, 2010.

[41] Bos Herbert Portokalidis Georgios. Sweetbait : Zero-hour worm detection and containment using low-and high-interaction honeypots. *Computer Networks*, 51(5) :1256–1274, 2007.

[42] Niels Provos. Developments of the honeyd virtual honeypot. `http://www.honeyd.org/`.

[43] Holz Thorsten Provos Niels. *Virtual honeypots : from botnet tracking to intrusion detection*. Pearson Education, 2007.

[44] Biondi Philippe Kaminsky Danielle Raynal Frederic, Berthier Yann. Honeypot forensics part 1 : analyzing the network. *IEEE Security & Privacy*, 2(4) :72–78, 2004.

[45] Kleinschmidt João Riboldi Jordao da Silva, Vargas Ivan. Capture and analysis of malicious traffic in voip environments using a low interaction honeypot. *IEEE Latin America Transactions*, 13 :777–783, 03 2015.

[46] McGrew Robert. Experiences with honeypot systems : Development, deployment, and analysis. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 9, pages 220a–220a. IEEE, 2006.

[47] Jonathan Rose. *Turning the tables : Loadable Kernel Module Rootkits deployed in a honeypot environment*. SANS Institute Information Security Reading Room, may 2003.

[48] Hota Chittaranjan Sadasivam Gokul Kannan. Scalable honeypot architecture for identifying malicious network activities. In *2015 international conference on emerging information technology and engineering solutions*, pages 27–31. IEEE, 2015.

[49] Yang T Andrew Sadasivam Karthik, Samudrala Banuprasad. Design of network security projects using honeypots. *Journal of Computing Sciences in Colleges*, 20(4) :282–293, 2005.

[50] Badreesh Shetty. Curse of dimensionality. https://towardsdatascience.com/curse-of-dimensionality-2092410f3d27.

[51] Singh Maninder Shukla Rohit. Pythonhoneymonkey : Detecting malicious web urls on client side honeypot systems. In *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, pages 1–5. IEEE, 2014.

[52] HoneyComb : Automated signature creation using honeypots. http://goo.gl/MepQu.

[53] Graphical Network Simulator-3. https://gns3.com/.

[54] Lance Spitzner. The honeynet projectthe honeynet project. https://www.honeynet.org/.

[55] Ziegler Robert Suehring Steve. *Linux Firewalls (Novell Press)*. Novell Press, 2005.

[56] Gao Yunhui Suo Xiangfeng, Han Xue. Research on the application of honeypot technology in intrusion detection system. In *2014 IEEE workshop on advanced research and technology in industry applications (WARTIA)*, pages 1030–1032. IEEE, 2014.

[57] CERT Insider Threat Team. Interesting insider threat statistics. http://goo.gl/DvAzB.

[58] The-Specter-Homepage. Specter project. http://www.specter.com/.

[59] UML-Homepage. The user-mode-linux kernel. http://user-modelinux.sourceforge.net/.

[60] vmware Homepage. Guide de l'utilisateur. http://www.vmware.com/-support/.

[61] Dr. Roy Winkelman. Chapter 1 : What is a network? https://fcit.usf.edu/network/chap1/chap1.htm.

[62] Manzano Yanet Yasinsac Alec. Honeytraps, a network forensic tool. In *Sixth Multi-Conference on Systemics, Cybernetics and Informatics*, 2002.